IMPACT OF THE BALANCE DISCOVERY ATTACK IN LIGHTNING NETWORK AND POSSIBLE COUNTERMEASURES

GIJS VAN DAM

UNIVERSITI KEBANGSAAN MALAYSIA

IMPACT OF THE BALANCE DISCOVERY ATTACK IN LIGHTNING NETWORK AND POSSIBLE COUNTERMEASURES

GIJS VAN DAM

THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

INSTITUTE OF IVI
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2024

KESAN SERANGAN PENEMUAN KESEIMBANGAN DALAM LIGHTNING NETWORK DAN KEMUNGKINAN LANGKAH-LANGKAH PENCEGAHAN

GIJS VAN DAM

TESIS YANG DIKEMUKAKAN UNTUK MEMPEROLEH IJAZAH
DOKTOR FALSAFAH

INSTITUT IVI
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2024

UNIVERSITI KEBANGSAAN MALAYSIA
The National University of Malaysia

PERAKUAN TESIS SARJANA / DOKTOR FALSAFAH
(DECLARATION OF MASTER / DOCTOR OF PHILOSOPHY THESIS)

A. MAKLUMAT PELAJAR/*STUDENT DETAILS*	
Nama *Name*	Gijs van Dam
No. Pendaftaran *Registration No.*	P95677
Fakulti/Institut *Faculty/Institute*	Institut Informatik Visual (IVI)
Program Pengajian *Program of Study*	~~Sarjana~~ / Doktor Falsafah | *~~Masters~~ / Doctor of Philosophy*
Tajuk Tesis *Thesis Title*	Impact of the Balance Discovery Attack in Lightning Network and Possible Countermeasures
Mel-e/*Email*: p95677@siswa.ukm.edu.my / gvandam@gmail.com	No. Telefon/*Tel.No: +31652403336*

B. PERAKUAN / *DECLARATION*

Merujuk kepada Dasar Harta Intelek UKM 2018, **tesis adalah hak milik UKM** seperti keterangan dibawah:

4.2.1 Harta Intelek yang direka cipta oleh pelajar termasuk tesis dan semua hasil penyelidikan ditulis oleh Pelajar UKM dalam tempoh pengajiannya di UKM direka cipta, dibangunkan atau dihasilkan menggunakan kemudahan, bahan, dana, atau sumber lain adalah hak milik UKM melainkan dinyatakan sebaliknya dalam Dasar ini atau dipersetujui sebaliknya oleh UKM dan Pelajar/Penaja.

*Referring to the UKM Intellectual Property Policy 2018, **the thesis is the property of UKM** as described below:*

4.2.1 Intellectual property created by the student, including the thesis and all research output produced by the UKM Student during their studies at UKM, including designed, developed or produced output using facilities, materials, funds, or other resources are the property of UKM unless otherwise stated in this Policy or otherwise agreed by UKM and the Student/Sponsor.

	RAHSIA ***CONFIDENTIAL***	Mengandungi maklumat rahsia di bawah AKTA RAHSIA RASMI 1972 *Consisting of classified information under the OFFICIAL SECRETS ACT 1972*
	TERHAD ***RESTRICTED***	Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan *Consisting of RESTRICTED information which has been determined by the organisation/body where the research was conducted*
X	**AKSES TERBUKA/ TIDAK TERHAD** ***OPEN ACCESS/ NON-RESTRICTED***	Saya membenarkan tesis ini diterbitkan secara akses terbuka, teks penuh atau dibuat salinan untuk tujuan pengajian, pembelajaran & penyelidikan sahaja *I give permission for this thesis to be published through open access, full text or copied only for study, learning, and research purposes.*

Bagi kategori Akses Terbuka/Tidak Terhad, saya membenarkan tesis (Sarjana/Doktor Falsafah) ini di simpan di Perpustakaan Universiti Kebangsaan Malaysia (UKM)* dengan syarat-syarat kegunaan seperti berikut:

For the Open Access/Non-Restricted category, I permit this (Master's/Doctoral) Thesis to be kept in the Universiti Kebangsaan Malaysia (UKM) Library with the following conditions:

1. Perpustakaan UKM mempunyai hak untuk membuat salinan untuk tujuan pengajian, pembelajaran, penyelidikan sahaja.

   UKM Library has the right to reproduce the thesis only for study, learning and research purposes.

2. Perpustakaan Universiti Kebangsaan Malaysia dibenarkan membuat satu (1) salinan tesis ini untuk tujuan pertukaran antara institusi pengajian tinggi dan mana-mana badan/ agensi kerajaan, tertakluk kepada terma dan syarat.

   UKM Library is allowed to make one (1) copy of this thesis for exchange purposes among higher education institutions and any government body/agency, subject to the terms and conditions.

Saya mengakui maklumat & tesis yang diberikan adalah benar & terkini. *I acknowledge the information & the thesis provided are correct & current.* Tandatangan Pelajar: *Student's Signature* *Tarikh / Date: 2 September 2024*	Saya memperakukan maklumat & tesis yang diberikan adalah benar & terkini. *I verify that the information & the thesis provided are correct & current.* Tandatangan Penyelia / *Supervisor's Signature*: Nama / *Name*: Cop Rasmi / *Official Stamp*: *Tarikh / Date:*

C. PENGESAHAN FAKULTI/INSTITUT | *VERIFICATION OF FACULTY/INSTITUTE*

Tandatangan/*Signature*:

Nama/*Name*:

Tarikh/*Date*:

Cop Rasmi/*Official Stamp*:

**Kuatkuasa:15 Julai 2021

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

August 15th, 2024

GIJS VAN DAM
P95677

ACKNOWLEDGEMENTS

For the major undertaking that has been the pursuit of this PhD I would like to express my gratitude first and foremost to my wife Marieke de Vries, without whom I would neither have started this PhD nor would I have finished it. Her academic experience, patience, understanding, and belief in my work have been foundational for me, enabling me to persevere through whatever challenge a PhD can throw at you. For this, and for so much more, I am deeply grateful.

I would like to extend my appreciation to my research supervisor, Dr. Rabiah Abdul Kadir and the other supervision committee members, Dr. Sharifah Md Yasin, Dr. Puteri Nor Ellyza Nohuddin and Prof. Emeritus Dato' Dr. Halimah Badioze Zaman. During the process of this research they were there to guide me and provide me with feedback.

Finally, I would like to thank my parents. Their unwavering support, and the values they instilled in me have been the guiding forces throughout my life.

IMPACT OF THE BALANCE DISCOVERY ATTACK IN LIGHTNING NETWORK AND POSSIBLE COUNTERMEASURES

ABSTRACT

Bitcoin has a low throughput of around 7 transactions per second. Payment Channel Networks (PCN) form a class of techniques created to solve the scalability problems that permissionless blockchains such as Bitcoin face. The Lightning Network (LN) is the biggest PCN to date. As of November 2023, LN is an interconnected network with 15 thousand nodes and 63 thousand channels, representing a total payment capacity of 5300 bitcoins (201M USD). The Lightning Network (LN) is a PCN that runs as a peer-to-peer network on top of Bitcoin and improves scalability by keeping most transactions off-chain without sacrificing the trustless character of Bitcoin. LN nodes establish trustless payment channels by locking funds in a funding transaction. The distribution of the funds can be modified with near-instant transactions without broadcasting them to the blockchain. By chaining channels together, it is possible for two nodes to transact without having a direct channel between them. This results in far fewer transactions needing to be broadcast and a theoretical number of off-chain transactions per second that can rival established, centralized payment systems. With off-chain transactions one can obtain more privacy than with on-chain transaction: individual payments through LN should be able to stay private. However, recent research suggests that this is not the case. Previous work has shown that channel balances can be tracked with the Balance Discovery Attack (BDA). The BDA has been proven to be successful in determining the balance of the largest part of channels in the network. Monitoring the changes in channel balances over time enables an adversary to track individual payments in LN as such the BDA is a big privacy risk. In this work we explore an extended algorithm for the BDA as well as a new type of attack that leverages the differences between LN client software implementations. Our improved algorithm improves the original BDA by performing two-way payments from both sides of a payment channel. The new type attack uses malformed payments to shutdown payment channels an adversary is not part of. Subsequently, this work applies approximate differential privacy as a possible countermeasure against the BDA by hiding payment amounts of a realistic size in LN. To our knowledge, this is the first time that strong privacy guarantees in the sense of approximate differential privacy are achieved in the setting of LN. We prove the feasibility with our prototype, compatible with LN today. Our solution is evaluated in terms of cost and utility and we show that the latter is not affected by our solution. We also introduce a second countermeasure, dubbed Payment Splitting and Switching (PSS): a way of splitting up payments in LN at intermediary hops along the payment path. PSS drastically reduces the information an attacker can obtain through a BDA. Using real-world data in an LN simulator we demonstrate that the information gain for the attacker drops up to 62% when PSS is deployed. Apart from its potential as mitigation against BDA, PSS also shows promise for increased LN throughput and as a mitigation against other types of attack.

IMPAK SERANGAN PENEMUAN BAKI DALAM RANGKAIAN KILAT DAN LANGKAH BALAS YANG BERKEMUNGKINAN

ABSTRAK

Bitcoin mempunyai daya pengeluaran yang rendah iaitu sekitar 7 transaksi sesaat. Saluran Pembayaran Rangkaian (PCN) membentuk teknik kelas yang dicipta untuk menyelesaikan masalah berskala blockchain tanpa kebenaran seperti muka Bitcoin. Rangkaian Kilat (LN) ialah PCN terbesar setakat ini. Sehingga November 2023, LN ialah rangkaian yang saling berkaitan dengan 15 ribu nod dan 63 ribu saluran, mewakili kapasiti pembayaran sebanyak 5300 bitcoin (201J USD). LN ialah PCN yang berfungsi sebagai rakan setara rangkaian di atas Bitcoin dan meningkatkan kebolehskalaan dengan mengekalkan kebanyakan urus niaga di luar rantaian tanpa mengorbankan ciri-ciri Bitcoin yang tidak dipercayai. Pengagihan dana boleh diubah suai dengan transaksi pantas tanpa menyiarkannya ke rantaian blok. Dengan merantai saluran bersama-sama, dua nod mungkin berurus niaga tanpa perlu menerusi saluran antara mereka. Ini menyebabkan sedikit transaksi yang perlu disiarkan dan bilangan teori urus niaga luar rantaian sesaat boleh menyaingi yang telah ditetapkan oleh sistem pembayaran berpusat. Dengan transaksi luar rantaian, seseorang boleh mendapatkan lebih privasi berbanding dengan transaksi dalam rantaian: pembayaran individu melalui LN sepatutnya boleh kekal persendirian. Walau bagaimanapun, penyelidikan baru-baru ini menunjukkan bahawa ini tidak berlaku. Kerja sebelumnya telah menunjukkan bahawa baki saluran boleh dijejaki dengan Balance Discovery Attack (BDA). BDA telah terbukti berjaya dalam menentukan keseimbangan yang terbesar sebahagian daripada saluran dalam rangkaian. Dengan memantau perubahan dalam baki saluran dari masa ke semasa, ini membolehkan pihak lawan menjejaki pembayaran individu dalam LN kerana BDA adalah privasi yang berisiko besar. Dalam kajian ini, kami meneroka algoritma untuk BDA serta jenis baharu serangan yang memanfaatkan perbezaan antara pelaksanaan perisian klien LN. Algoritma kami dipertingkatkan dengan menambah baik BDA asal di mana melakukan pembayaran dua hala pada kedua belah saluran pembayaran. Serangan jenis baharu menggunakan pembayaran yang salah bentuk untuk menutup saluran pembayaran yang bukan sebahagian daripada musuh. Selepas itu, kajian ini terpakai dengan menganggarkan kemungkinan privasi pembezaan sebagai langkah balas terhadap BDA dengan menyembunyikan jumlah pembayaran bersaiz realistik dalam LN. Ini merupakan kali pertama jaminan privasi yang kukuh, dalam erti kata privasi pembezaan anggaran dicapai dalam tetapan LN. Kajian membuktikan kebolehlaksanaan prototaip ini serasi dengan LN pada hari ini. Penyelesaian ini dinilai dari segi kos dan utiliti, dan ia menunjukkan bahawa kaedah yang kedua tidak dipengaruhi oleh penyelesaian ini. Kajian ini juga memperkenalkan langkah balas kedua, yang digelar sebagai Bayaran Pemisahan dan Penukaran (PSS) iaitu membahagikan pembayaran dalam LN di pengantara melompat di sepanjang laluan pembayaran. PSS secara mengurangkan maklumat penyerang yang boleh diperolehi melalui BDA. Menggunakan data dunia sebenar dalam simulator LN, kajian ini menunjukkannya keuntungan maklumat untuk penyerang menurun sehingga 62% apabila PSS digunakan. Selain daripada potensinya sebagai mitigasi terhadap BDA, PSS juga menjanjikan peningkatan LN pengeluaran adalah sebagai mitigasi terhadap jenis serangan lain.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 OVERVIEW

Bitcoin is the cryptocurrency with the largest market capitalization. In november 2021 Bitcoin hit an all-time high market cap of 1.21 trillion US Dollar. It is still by far the most popular cryptocurrency. With this popularity some of its flaws have also risen to the surface, scalability being one of them. Several solutions have been proposed to solve these scalability issues, but only one of them has been implemented so far: the Lightning Network (LN). This network runs on top of the Bitcoin blockchain and has the theoretical property of increasing the throughput of transactions dramatically. These solutions have the added benefit of increasing the level of anonymity of the participants. Until now the privacy of the participants in LN have not been formally analyzed. This study seeks to analyze and measure the privacy that can be obtained by participants in the Lightning Network while being exposed to known and yet to be developed attacks on privacy.

1.2 RESEARCH BACKGROUND

Bitcoin (Nakamoto 2008) is a fully decentralized cryptocurrency that can operate in a trustless environment. Bitcoin was the first cryptocurrency that solved the Byzantine Generals Problem (Lamport, Shostak, and Pease 1982) in the *specific context* of cryptocurrency, by incentivizing the nodes in a network to behave honest, and by embracing randomness. The latter meaning that there's a Bayesian approach to the confirmation of a transaction: the certainty of a transaction approaches 100% asymptotically over time.

Because a decentralized cryptocurrency operates without a trusted, central party, participants need to have a way to trust the protocol without relying on another party. This is primarily achieved by broadcasting *all* transactions transparently on to the blockchain. This gives participants in the Bitcoin network the opportunity to download the entire blockchain to verify that transactions are valid, and that the sender of a payment of which they are the receiver isn't trying to double spend their coins.

New transactions in Bitcoin are shared with other nodes in a peer-to-peer network through a gossip protocol (Birman 2007). Those, still unconfirmed, transactions are stored in the mempool. The term "mempool" is made out of two words: "memory" and "pool", and indicates a storage for pending transactions that are yet to be confirmed. Every 10 minutes a random node, chosen through a proof-of-work lottery, generates a new block of transactions. This process of doing the necessary proof-of-work to win the lottery and creating a new block of transactions is called mining. A transaction is said to be confirmed if it is part of a block on the blockchain. But a new block doesn't necessarily point to the latest block. Under certain conditions it can point to an earlier block, forking the blockchain. If a bifurcation like this happens, the Bitcoin protocol prescribes that the longest branch is the valid one. So new blocks should point to the latest block in the longest branch. The blocks in short branches are then considered to be orphaned. Any transactions in it should not be considered valid transactions. It is because of this that it is safer to wait for a number of blocks, to get a higher degree of certainty that a transaction is final. This is called waiting for confirmations and each block that extends the branch that contains the transaction (including the block containing the transaction itself) is considered a confirmation. most sources recommend to wait for six confirmations as a rule of thumb. There is nothing magical about that number, but it seems to be a nice trade-off between not waiting too long (6 confirmations is roughly equal to an hour) and getting a level of certainty that is acceptable. The certainty of a transaction is a function of the attacker's hashrate (as a proportion of the total network hashrate) and the number of confirmations. If we assume the attacker controls 10% of the hashrate and a transaction has six confirmations the certainty of a transaction is around 99,9% (Rosenfeld 2014).

A block of transactions points to its predecessor by means of a hash pointer,

resulting in a chain of blocks that gets extended with a new block every 10 minutes. The resulting data structure is called a blockchain. The size of a single block in Bitcoin is limited to 1MB. With a basic transaction taking up 250 bytes and an average transaction size of 500 bytes the network has a maximum throughput of 4000 transactions per block and an average throughput of 2000 transactions per block. This boils down to 3-7 transactions per second.

Increasing the throughput by either increasing the block size or the rate at which blocks are generated reduces the security of the Bitcoin network (Sompolinsky and Zohar 2015). However, increasing the scalability of Bitcoin (without abandoning security) remains desirable. Firstly because if the amount of transactions being broadcast exceeds the capacity of the network, the law of supply and demand dictates that the transaction fees will increase (Easley, O'Hara, and Basu 2019) Secondly, if we want to achieve a viable alternative to current centralized payment networks we need to achieve comparable throughput which is in the order of magnitude of several thousand transactions per second.

The transparency of all transactions in a public blockchain means that privacy and anonymity can only be achieved by separating the transaction and the identities of the participants performing the transaction. This separation of transaction and identity is obtained by asymmetric encryption where the hash of the public key acts as an account number or address. These addresses act as pseudonyms for the participant. The anonymity that is achieved by this separation knows its limits. The system of pseudonymity in Bitcoin has been the basis for a lot of prior research (Meiklejohn et al. 2013; Garcia-Alfaro et al. 2015; Langenheldt et al. 2018).

The limitations with regards to privacy and capacity described above provided the incentive for several improvement proposals. These proposals can be categorized into three different categories:

- Alt-coins, new cryptocurrencies that are completely separate from Bitcoin
- Protocol improvements, improvements to the Bitcoin protocol itself
- Second layer improvements, overlays that run on top of the Bitcoin blockchain and don't require any or only small modifications to the Bitcoin protocol

Payment Channel Networks (PCNs) are an example of the latter type of improvement. PCNs improve both privacy and capacity and as such hit two birds with one stone. PCNs achieve higher throughput by keeping transactions off-chain, meaning they don't get broadcast unto the blockchain. This is made possible by opening up payment channels between participants, and keep private off-chain balances of transactions between any two participants. Upon closing of a payment channel the final balance is broadcast to the Bitcoin blockchain. This way the amount of transactions that need to be broadcast is drastically reduced. Theoretically this could lead to throughputs of PCNs that rival those of VISA and other global payment networks. Lightning Network (LN) (Poon and Dryja 2016) is the first PCN that has been put into production and runs on top of Bitcoin.

1.2.1 Lightning Network

At its core LN allows for participants to pay each other. Central to LN, and its raison d'être, is its property that individual transactions do not have to be broadcast to the blockchain. Participating in LN requires operating an LN *node*. Each LN node is identifiable through a public key, the *node id*. Nodes connect to each other using the node id in a peer-to-peer (P2P) network. An LN node communicates with a bitcoin node to broadcast transactions or retrieve information on broadcast transactions.

After an LN node connects with a peer through the P2P network it can initiate the opening of a *channel* between them. After agreeing on the channel parameters, the two parties deposit bitcoins in a 2-of-2 *multi-signature* address. The deposits reflect the initial distribution of the funds. This transaction is called the *funding transaction*. The total of the deposits reflects the channel's *capacity*, i.e., the maximum transferrable amount. Currently, LN channels are single-funded; the initiator for the opening provides all the funding. The protocol for dual-funded channels, where both nodes contribute funds to the funding transaction, is currently still in draft[^1]. After a funding transaction gets enough confirmations, the channel is open.

Once a channel is open, the two nodes connected by it can create transactions in both directions if their local balance allows for it. A single-funded channel initially

only allows for a payment in one direction since the balance is completely one-sided. Every *LN transaction* leads to updated balances that reflect the current state of the channel. Each state is reflected in a *commitment transaction* — a bitcoin transaction that spends the funding transaction and divvies up the funds according to the latest state. Commitment transactions are valid bitcoin transactions that could, but normally would not be broadcasted to the blockchain. Only when one of the channel partners unilaterally, or both cooperatively decide to close the channel, is the final state broadcasted to the blockchain.

The outputs in a commitment transaction related to payments that are being processed are called *hash time-locked contracts* (HTLCs). The output encumbered with an HTLC is spendable for one party by providing the pre-image of a given hash or for the other party after a timeout. Imagine Alice wants to buy something from Bob for an agreed amount x. Bob generates a random number r (the pre-image) and sends the *payment hash* $H(r)$ to Alice in a message called an *invoice* over the P2P network. Alice now offers Bob an HTLC for x coins, spendable for him before the timeout, provided he knows the pre-image of $H(r)$. Bob can now claim the payment since he knows r. If he does not know r, or if the amount does not meet the agreed amount, Bob will let the timeout expire so that Alice can reclaim her payment. In both cases, the HTLC is considered *resolved* and the outputs in the next commitment transaction will reflect the updated balances in its outputs. At any time a payment channel can track multiple unresolved HTLCs.

LN uses source-based routing. This means that the payer is responsible for finding a route to the payee. HTLCs make it possible to create multi-hop routes that include nodes that are not the payer nor the payee. Consider Alice again, but now she wants to buy something from Charlie. Alice does not have a payment channel with Charlie, but based on her local view of the network graph she can find a route to Charlie through Bob. Charlie creates an invoice for Alice to pay. Alice offers an HTLC to Bob, who in turn offers an HTLC to Charlie based on the same payment hash but with a somewhat smaller timeout. Charlie, knowing the pre-image, can now claim Bob's HTLC. By doing so Charlie reveals the pre-image to Bob, who has enough time to claim Alice's HTLC since the timeout on that HTLC is slightly longer. HTLCs make multi-hop payments

in LN *atomic*: either they succeed completely, or they do not succeed at all. Routing payments is incentivized by routing fees that are collected by the relaying nodes, known as *hops*. The routing fee expresses itself as a small difference between the HTLC offered to a hop and the HTLC that the hop offers to the next node in the route. The payer needs to take these routing fees into account when creating a route.

Source-based routing demands that the payer is aware of the network graph to be able to find a route to a node with whom the payer has no direct channel. LN uses a gossip protocol for announcing new nodes and channels. Channels that are announced in this way are public, but two nodes can also decide to refrain from announcing their channel in which case the channel is considered private. It is important to note that the scalability benefits of LN come from the users not having to broadcast all their payments to the blockchain (Malavolta et al. 2019). This is true for direct payments between two users, but the benefit is amplified through multi-hop payments. So for the scalability benefits to fully materialize, LN needs a graph with a substantial amount of public channels.

To protect privacy between the relaying nodes, LN uses an *onion routing* scheme (Reed, Syverson, and Goldschlag 1998) based on the Sphinx Mix format (Danezis and Goldberg 2009). This ensures that each node on a routing path can only identify its immediate predecessor and successor.

A node might be feel tempted to unilaterally close a channel by broadcasting an old state that is beneficial for it compared to the current state. To prevent this from happening LN uses a *penalty* system. The outputs of a commitment transaction that pay out to the holder of the commitment transaction are spendable after a timeout or can directly be spent by the channel partner with the *commitment revocation private key*. It follows that the states held by both channel partners are asymmetric: only outputs that pay to yourself are encumbered with the revocation mechanism. Every time the channel partners agree on a new state, they exchange the commitment revocation private keys of the previous state. So, at any time, a node holds all commitment revocation private keys for all commitment transactions its channel partner holds, *except* for the current state. When a node broadcasts an old state, its channel partner can dispute this before the timeout and claim all channel funds.

The improved anonymity of LN is comparable to the anonymity achieved improvements that have the sole goal to increase anonymity, like Mixcoin, Blindcoin and TumbleBit (Conti et al. 2018). However, the lack of a rigorous definition of the PCN protocols, the absence of a threat model, and the ambivalent interpretations of the concept of anonymity have hindered formal analysis of privacy in the context of PCN's (Malavolta, Moreno-Sanchez, Kate, Maffei, and Ravi 2017).

1.2.2 Threat Model & Value Privacy

In recent years a common threat model has been formalized which has been the foundation of most of the privacy analysis research (Malavolta, Moreno-Sanchez, Kate, Maffei, and Ravi 2017). This threat model has four notions of interest:

- Balance security: participants should not run the risk of losing coins to a malevolent adversary.
- Serializability: executions of a PCN are serializable as understood in concurrency control of transaction processing, i.e. for every concurrent processing of payments there exists an equivalent sequential execution.
- (Off-path) value privacy: malicious participants in the network cannot learn information about payments they are not part of.
- (On-path) relationship anonymity: given at least one honest intermediary, corrupted intermediaries cannot determine the sender and the receiver of a transaction better than just by guessing.

In our work we focus mainly on the notion of (Off-path) value privacy.

1.2.3 Onion Routing with Sphinx

LN uses onion routing (Reed, Syverson, and Goldschlag 1998) to ensure privacy along a payment path. The encryption scheme employed by LN is called Sphinx (Danezis and Goldberg 2009). For each node in the payment path, the Sender creates a shared secret.

The Sender starts by creating a session key, a random 256 bits bitstring. This session key *is* the first ephemeral secret key. The Sender then shares the first ephemeral *public* key with the first node along the payment path. It does so by sending it unencrypted in the onion header. Now both the Sender and the first node can create the first shared secret key using Elliptic Curve Diffie-Hellman (ECDH). To this end, the Sender uses the first ephemeral secret key and the `node_id` of the first node, which is its public key. The first node uses its secret key and the ephemeral public key it received from the Sender.

For the next node, the Sender creates a new ephemeral secret key. It does so by creating a tweak factor. This tweak factor is the hash of the concatenation of the first ephemeral public key and the first shared secret. Multiplying the tweak factor with the first ephemeral secret key gives the second ephemeral secret key. The Sender shares the second ephemeral *public* key with the second node in the payment path, again unencrypted in the onion header. Now both the Sender and the second node can create the second shared secret key. This process is repeated for each node along the payment path.

The Sender now creates the onion, encrypting each layer of it using the shared secret keys. The outermost layer is encrypted using the first shared secret key, the second layer using the second shared secret key and so on. This encapsulation in layers of encryption achieves that each node along a payment path can only "peel" away a single layer, exposing the next destination. So each node is only aware of the node from which it got the onion and the node to which the remaining layers of the onion have to be sent.

1.2.4 Balance Discovery Attack

One of the vulnerabilities that have been found in LN is the Balance Discovery Attack (BDA). Herrera-Joancomartí et al. (2019) introduced a technique to reveal a channel's balance in the Lightning Network. This attack capitalizes on the inherent trial-and-error process involved in identifying a viable payment route in LN. The challenge is for the payer to determine a path to the payee that has enough balance at each node to facilitate the payment. The BDA operates by initiating payments that are deliberately

set up to fail using random, invalid payment hashes. The payment will fail by definition (since it uses a fake payment hash), but the error message can be used to obtain extra information. When a payment does not go through because a node along the path does not have sufficient funds, it returns an error message such as `InsufficientFunds` or `TemporaryChannelFailure`. Conversely, if a payment fails at the payee because the payment hash is unrecognized (yielding an `UnknownPaymentHash` error), it implies that all nodes up to the recipient had enough funds to relay the payment.

An adversary can use these error messages to perform a binary search, consecutively adjusting the payment amount to zero in on the exact balance of a channel, accurate down to a single millisatoshi, the smallest payment increment of Bitcoin. The binary search continues with this method, using the error messages to adjust the bounds of the search until the channel balance is determined within the desired accuracy.

The BDA poses a significant privacy risk as it allows an adversary to track channel balances over time, which could be used to trace transaction paths and transaction amounts through the network.

1.3 PROBLEM STATEMENT

The level of anonymity to be attained in Bitcoin transactions is well studied (Spagnuolo, Maggi, and Zanero 2014; Koshy, Koshy, and McDaniel 2014; Barber et al. 2012; Androulaki et al. 2013; Meiklejohn and Orlandi 2015; Meiklejohn et al. 2013; Reid and Harrigan 2013). but the amount of anonymity to be attained in Lightning Network (LN) running on top of Bitcoin, has yet to be determined through formal privacy analysis. Since in LN most transactions are not broadcasted to the blockchain, LN is expected to provide better privacy to its users when compared to Bitcoin [@Malavolta2017]. Despite this, LN introduces an additional and unknown attack surface. While much of the existing research on the LN has concentrated on scalability improvements (Gangwal, Gangavalli, and Thirupathi 2023; Rebello et al. 2024; Sguanci, Spatafora, and Vergani 2021), there is a significant need to address the privacy and anonymity claims of LN.

Earlier work on the formal privacy analysis in LN was hindered by a lack of a rigorous definition of privacy in LN. Based on the threat model put forward by Malavolta, Moreno-Sanchez, Kate, Maffei, and Ravi (2017), two privacy notions have been identified: (Off-path) value privacy and (On-path) relationship anonymity. This study focuses on off-path value privacy (Malavolta, Moreno-Sanchez, Kate, Maffei, and Ravi 2017). Intermediate nodes on a payments path can discern the value of the payment, if they factor in transaction fees. However, nodes that are not on the payment path should remain oblivious to the precise transaction amount. This is why value privacy is defined as *off-path* value privacy, and wherever off-path is not specifically mentioned it should be assumed.

The specific gap addressed by this study is the lack of formal analysis and understanding of off-path value privacy in LN. Previous research has not thoroughly investigated the extent to which off-path value privacy is maintained in LN, nor has it explored the impact of potential attacks, such as the Balance Discovery Attack (BDA), on this privacy aspect. This study aims to fill this gap by evaluating the viability of off-path value privacy in LN and assessing the risks associated with BDA as well as potential mitigations against BDA.

In contrast to earlier work, which has primarily focused on mitigating other types of attacks such as griefing (Mazumdar, Banerjee, and Ruj 2020), jamming (Shikhelman and Tikhomirov 2022) , systemic attacks (Harris and Zohar 2020) or studies off-path value privacy without exploring mitigations (Kappos et al. 2021), this study specifically addresses the gap in research on mitigating the Balance Discovery Attack (BDA). We aim to extend the understanding of BDA and develop strategies to mitigate this specific type of attack (See figure 1.1).

Figure 1.1 Extending BDA and proposing mitigations

Given LN's role in addressing Bitcoin's scalability issues, understanding and

mitigating the risks to value privacy in LN is crucial. This research will contribute to a more comprehensive understanding of LN's privacy guarantees and help in developing strategies to enhance user anonymity.

1.4 RESEARCH OBJECTIVES

This research aims to study whether LN offers off-path value privacy by employing known and new attacks. Subsequently this study aims to propose mitigations against said attacks and quantify the privacy improvements that those mitigations offer.

To achieve these aims, the following objectives are set:

1. Evaluate the effectiveness of the balance discovery attack (BDA) in compromising off-path value privacy in LN, building on the work of Herrera-Joancomarti (2019)
2. Develop and propose novel mitigation strategies to counteract the balance discovery attack in LN
3. Quantify the privacy impact of BDAs on LN by measuring privacy loss in terms of ε-differential privacy (Dwork 2006), information gain (Biryukov, Naumenko, and Tikhomirov 2022) and channel information coefficient (Tikhomirov et al. 2020).

1.5 RESEARCH SCOPE

This research focuses on off-path value privacy with the aim of both exploring and evaluating the impact of the BDA and at the same time provide possible mitigations.

The scope of this study will be value privacy as defined by Malavolta (Malavolta, Moreno-Sanchez, Kate, Maffei, and Ravi 2017) with regards to the protocol of Lightning Network running on Bitcoin mainnet. The messages and packets routed through this network are defined in the BOLT specifications ("Basis of Lightning Technology (BoLT), Lightning Network In-Progress Specifications" 2022).

The decision to focus exclusively on off-path value privacy stems from the unique challenges and vulnerabilities associated with off-path transactions within the Lightning Network. Off-path value privacy refers to the protection of transaction values that are not directly involved in the payment path but could be inferred through indirect means.

In line with our focus on off-path value privacy, from the perspective of the malicious actor we will focus on the BDA. This decision is rooted in several key considerations:

1. BDA directly threatens the privacy and security of the Lightning Network by potentially exposing channel balances. This type of attack has immediate and significant implications for user privacy, network security, and overall trust in the Lightning Network. Addressing BDA is therefore of paramount importance for the integrity of the network.
2. By concentrating on BDA, the research can delve deeply into the mechanics, implications, and mitigation strategies for this specific attack. A focused approach allows for a more thorough analysis and a detailed understanding of BDA, which is necessary for developing effective countermeasures.
3. There is already a significant body of research on various types of attacks however, BDA remains relatively underexplored.

We do not assume a global passive adversary, but instead, we assume a computationally efficient attacker who can run some fraction of the nodes in LN. Those nodes can deviate from the protocol in any way the attacker sees fit, but we assume all other nodes run LN software without bugs and do not leak information through side-channels.

Where applicable we assume that the attacker is computationally bounded (i.e., probabilistic, polynomial time) and can not break the underlying cryptographic primitives.

1.6 SIGNIFICANCE OF STUDY

> "Promoting greater transparency in the use of digital currencies serves to protect the integrity of the financial system and strengthen incentives to

prevent their abuse for illegal activities

Members of the public are therefore advised to undertake the necessary due diligence and assessment of risks involved in dealing in digital currencies or with entities providing services associated with digital currencies."

Source: Bank Negara - Policy document on Anti-Money Laundering and Counter Financing of Terrorism (AML/CFT) – Digital Currencies (Sector 6)

The above quotes from Bank Negara give a backdrop to the significance of this study. The results will provide more transparency in the use of digital currencies by supplying new techniques for analyzing the blockchain and Lightning Network especially and assess the risks involved in dealing with digital currencies through LN.

In general the privacy of personal information and financial information in particular is an important area of focus in today's digital world. This work adds to the growing foundation of privacy research of LN and other Layer-2 solutions.

Enhancing off-path value privacy has practical implications for the design and implementation of more secure and private payment channels. This focus can lead to the development of improved protocols and mechanisms that, enabling a future where citizens are less vulnerable to attacks on their privacy and funds.

1.7 RESEARCH CONTRIBUTIONS

The main contributions of this study are an algorithm for an extended BDA and two different types of mitigations against BDAs that both can be deployed in LN as it functions today.

The extended BDA algorithm is able to disclose the balance of 98.4% of all channels.

The first type of mitigation applies approximate differential privacy to hide payment amounts of a realistic size in LN. This is the first time that strong privacy

guarantees in the sense of approximate differential privacy are achieved in the setting of LN.

The second type of mitigation applies the concept of intermediary payment splitting to en-route payments, and explores the effect on the obtainable information gain for the attacker. This is to date the strongest mitigation against BDA in terms of prevented information gain.

Other contributions include:

- A novel attack that enables an attacker to close down remote channels.
- A Core Lightning plugin for intermediary payment splitting.
- Several contributions to open source project such as Simverse and LN Probing Simulator

1.8 ORGANIZATION OF THESIS

This thesis is divided into six chapters: The first chapter is the introduction which is the current chapter. The second chapter is the literature review which provides an overview of the previously published literature on LN and its privacy and security properties in particular. The following chapter describes the methodology by specifying the approach and rationale for this thesis. It is followed by the results in chapter four, laying out our main findings. Chapter five is the discussion in which we evaluate our findings and see how they relate to previous work and our research objectives. We finish with the conclusion and future work, where we summarize our research and lay out implications and insights as well as pointing into the direction of possible avenues for future research. Chapter 1 and 2 explain the theoretical framework of this research, which is the foundation of the work which is detailed in chapter 3 and 4. Figure 1.2 shows the structure of this thesis.

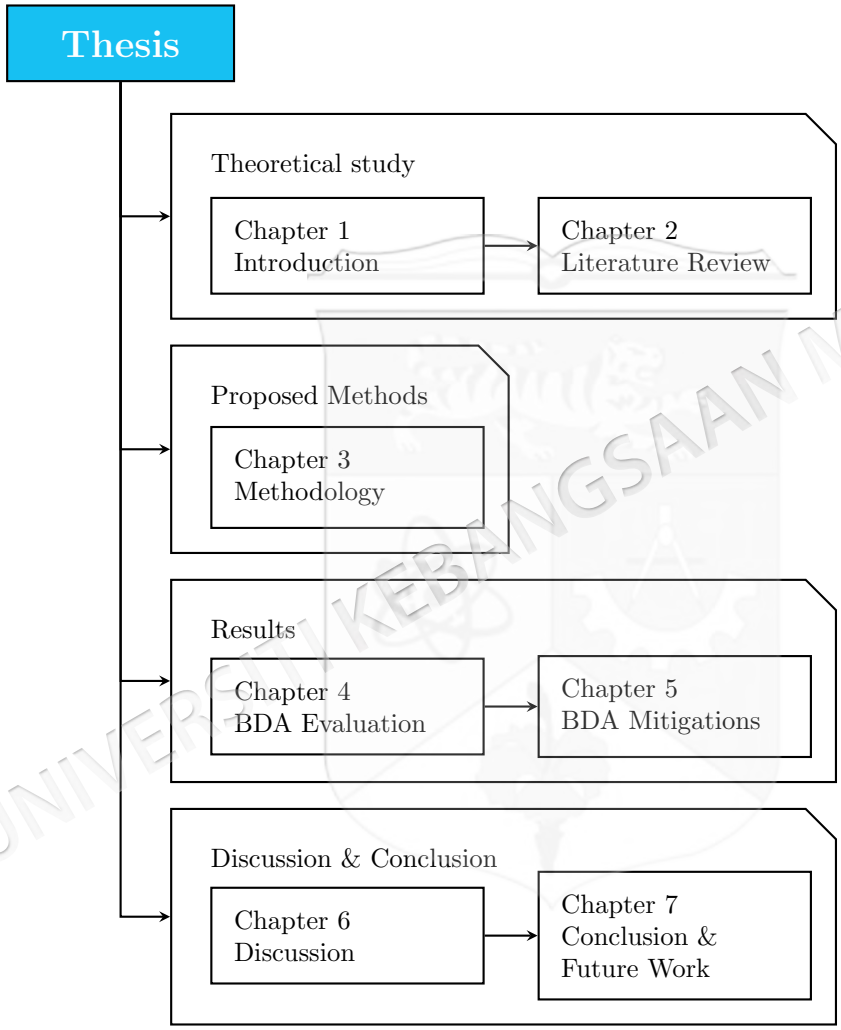

Figure 1.2 Thesis organization

CHAPTER II

LITERATURE REVIEW

2.1 OVERVIEW

This chapter summarizes the relevant academic literature on the subject of blockchain technology and the scalability problems that these technologies face. Layer-2 solutions are a class of technology intended to solve these problems, and in this chapter we give an overview of the relevant Layer-2 applications for different cryptocurrencies before we focus on LN, its challenges and in particular its privacy and security properties. The objective is to comprehensively review the literature and to provide a backdrop for our research, highlighting the gaps in the current literature and identifying the research issues.

The structure of this chapter is as follows. We start with an overview of the Layer-2 solutions chapter 2.2 for different cryptocurrencies. Next, an overview of LN challenges is given in chapter 2.3. Subsequently chapter 2.4, chapter 2.5 and chapter 2.6 go into detail on the specific challenges divided into three categories: Transaction security, network robustness and privacy. chapter 2.6 then dives further into the specific aspects of LN privacy, and focusses on balance privacy (chapter 2.6.3) and lays out the Balance Discovery Attack. It explores the proposed countermeasures and the potential countermeasures of which this thesis is an elaboration. Finally chapter 2.7 gives summarizes and concludes this chapter.

2.2 LAYER-2 SOLUTIONS

Blockchain technology, despite its transformative potential, is significantly constrained by scalability issues, which adversely affect its performance and broader adoption. To address these limitations, Layer-2 solutions have emerged as a critical area of development, offering various approaches to enhance blockchain scalability without compromising security and decentralization (Sguanci, Spatafora, and Vergani 2021). This distinct class of solutions known as Layer-2 protocols tries to address blockchain scalability issues through innovative approaches. Layer-2 protocols enhance transaction processing rates, reduce latency, and lower fees by minimizing reliance on the underlying, slower, and more costly main blockchains. In these systems, the main chain primarily serves as a tool for establishing trust and resolving disputes among Layer-2 participants, with only a minimal number of transactions being recorded on the main chain. Consequently, Layer-2 blockchain protocols hold significant potential to revolutionize the field (Gangwal, Gangavalli, and Thirupathi 2023).

These solutions can be broadly categorized into several classes based on their underlying mechanisms and approaches, leading to the following taxonomy of such protocols and implementations figure 2.1.

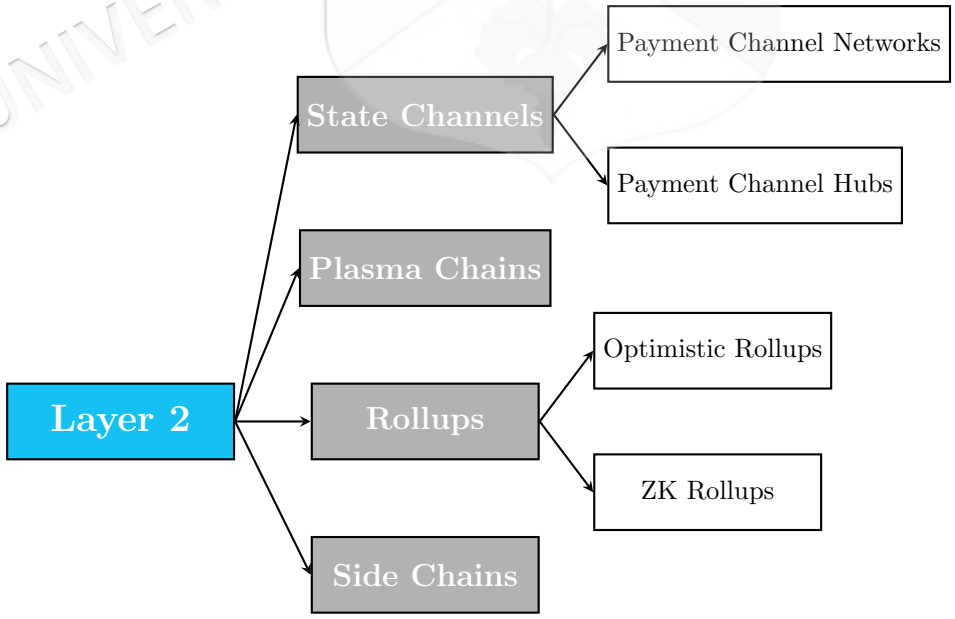

Figure 2.1 A taxonomy of Layer-2 solutions and protocols

2.2.1 State Channels

A state channel (Miller et al. 2019) is a mechanism that enables the exchange or transfer of states between two or more participants. These states can represent various applications, such as voting or auctions. Typically, a state channel is built using threshold signatures, commonly known as multisig, and timelock instructions. Participants sign a multisig contract and lock funds to engage in the transfer. State channels are typically established using a smart contract. Within these channels, states are exchanged among all participants. Once all transactions are completed, the participants commit the final state of the channel to the main chain.

Such a channel is particularly advantageous for frequent state exchanges between participants, as off-chain exchanges are much faster than on-chain exchanges. Therefore, state channels significantly improve the slow transaction rate of the parent blockchain.

a. Payment Channel Networks

One of the primary scalability goals for blockchains has been to support (micro-)payments with near-instant confirmation, and reduced fees (Dmitrienko, Noack, and Yung 2017; Hu and Zhang 2018; Pass and Shelat 2015). Payment channels are a bespoke form of state channels specifically for payment-specific applications. Initially designed for one-way payments (Hearn 2013), payment channels have evolved into bi-directional channels (Decker and Wattenhofer 2015; Poon and Dryja 2016), allowing each participant to both send and receive payments. The Lightning Network (Poon and Dryja 2016) is such a bi-directional payment channel network designed to facilitate fast, low-cost transactions by enabling off-chain exchanges between participants. Raiden (Network-Fast 2018) is a payment channel network similar to the Lightning Network, but it is designed specifically for the Ethereum blockchain, whereas the Lightning Network is built for Bitcoin. The challenges to Lightning Network will be expanded upon further in this literature review in section 2.3.

b. Payment Channel Hubs

Payment Channel Hubs introduce a specialized node known as a hub, which serves as the central point in a star topology and facilitates payments to connected nodes. This strategy aims to minimize routing overhead and the funds locked by individual nodes in Payment Channel Networks (Avarikioti et al. 2018). By using multiple interconnected hubs, the network can achieve shorter routing paths, thereby reducing both routing and collateral costs for each channel. However, as the number of channels and transaction volume increase, the total funds a hub must lock can become substantial. This issue is exacerbated when transactions predominantly flow in one direction, leading to expensive and slow rebalancing operations (Khalil and Gervais 2017).

2.2.2 Plasma Chains

Plasma chains (Poon and Buterin 2017) are a framework for creating a hierarchy of child chains that are anchored to the main blockchain. These child chains can process transactions independently and periodically submit a summary to the main chain, enhancing scalability and reducing congestion.

The Plasma chain introduces a UTXO-based ledger system operating atop an account-based blockchain, enabling multiple blockchains to function as branches of a tree through a series of smart contracts. Each branch, or blockchain, can have sub-branches, known as child chains, with each Plasma chain maintaining its own block validation mechanism independent of its parent chain. Despite this independence, all computations across the chain hierarchy are globally enforced by a single root chain. However, Plasma chains face challenges (Bez, Fornari, and Vardanega 2019) such as increasing data storage costs, high computational demands, and a lack of native support for instant finality.

2.2.3 Rollups

Rollups are non-custodial side chain solutions aimed at alleviating the load on the Layer-1 chain by using data compression techniques and smart contracts. Unlike Plasma chains, Rollups keep minimal data on-chain. A Merkle root is stored on-chain, to enable verification. Transactions are processed off-chain in batches and then bundled for on-chain verification. The smart contract holds the Merkle root (state root) on-chain, reflecting the current state (e.g. individual balances) of the Rollup, which can be derived and verified from the on-chain data. To conserve space, the Merkle tree itself is not stored on-chain. After each batch of transactions, a new state root is computed. Anyone can submit a batch by including the compressed transactions, the previous state root, and the new state root. If the current state root in the contract matches the previous state root in the new batch, the contract updates its state root to the new one. For batches needing external inputs or outputs, the required funds are transferred to the contract before processing or sent to outputs after processing the transactions.

Rollups employ compression techniques to minimize the on-chain transaction footprint, thereby conserving space and enhancing the scalability of the Layer-1 chain. Additionally, they effectively address the data availability issue (Huang et al. 2024) through the use of fraud proofs.

a. Optimistic Rollups

Optimistic Rollups adopt an optimistic strategy, presuming transactions are valid unless contested (Thibault, Sarry, and Hafid 2022). This approach eliminates the need for default verification computation, thereby enhancing scalability. The contract, however, keeps a record of state root updates and associated batch hashes. To challenge a batch, one must submit a proof of incorrect computations, known as fraud proofs, on-chain. The contract then verifies this proof. If the proof is validated, the contract reverses the erroneous batch and all subsequent batches.

b. zk Rollups

Unlike Optimistic Rollups, zk Rollups treat every transaction with suspicion. Each batch includes a cryptographic proof, known as a validity proof, which verifies that the new state root correctly reflects the execution of the batch of transactions. These proofs are generated using zk-SNARK and PLONK protocols (Gabizon, Williamson, and Ciobotaru 2019). Although the creation of validity proofs is computationally intensive, their verification on-chain is swift.

2.2.4 Side Chains

A Side Chain (Back et al., n.d.; Singh et al. 2020) is an autonomous distributed ledger operating alongside the main chain. Its primary objectives are to alleviate the main chain's load by offloading computationally intensive tasks and to facilitate asset transfers across different blockchains. Side chains typically employ their own consensus mechanisms (e.g., proof-of-authority, proof-of-stake) for transaction processing. They communicate and exchange funds with the main chain through a two-way bridge known as a two-way peg. The effectiveness of a side chain hinges on its ability to rapidly exchange information with the main chain and efficiently process transactions, often utilizing custom block parameters for this purpose.

2.3 CHALLENGES OF LIGHTNING NETWORK

Lightning Network has seen its usage increase drastically over the past few years. The use of Lightning Network is estimated to have doubled in 2021, while the value of payments has increased by over 400% ("The State of Lightning: Volume 2" 2022). With its rise in popularity also its problems have been exposed. These problems can be divided into three categories (Zhao et al. 2021): Transaction security, network robustness and privacy. Those three categories can be subdivided according to figure 2.2

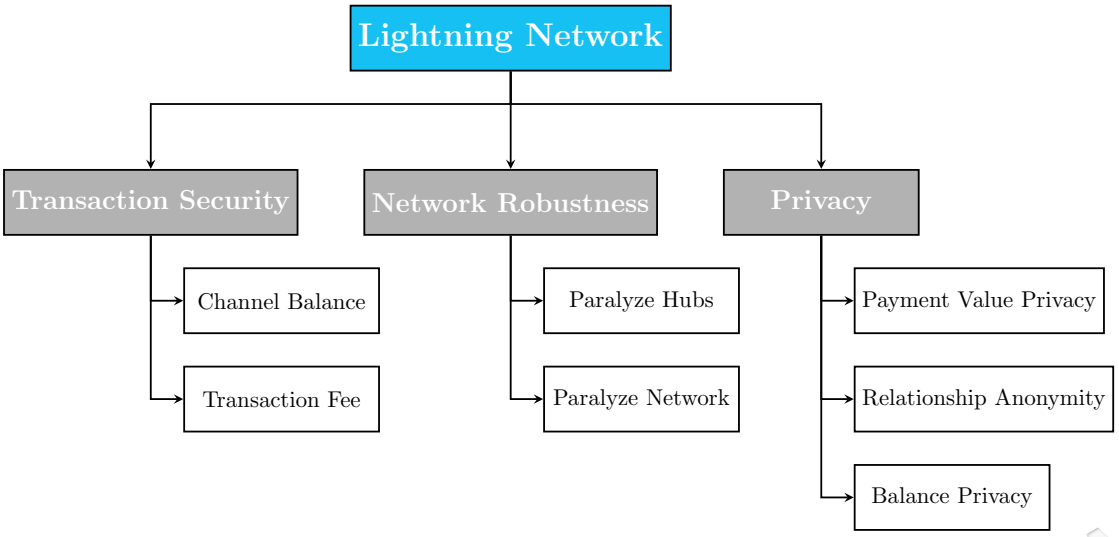

Figure 2.2 A systematic overview of the LN challenges

2.4 TRANSACTION SECURITY

LN is used for the transfer of money. Transaction security is probably the single most important requirement for any payment network. Within the context of LN, transaction security refers to two things: Deposit security and transaction fee security.

2.4.1 Deposit security

In chapter 1.2.1 we explained the penalty system that is in place to prevent a malicious user from broadcasting a stale commitment transaction. When this happens the counterparty has a limited time frame to broadcast the penalty transaction that penalizes the malicious user. However if the maligned party does not come online within said time frame, the opportunity for broadcasting the penalty transaction will pass, resulting in a loss of funds.

To address the perpetual connectivity challenge, the idea of delegating arbitration (Tadge Dryja 2016) to entities termed as “watchtowers” has been suggested. Such watchtowers are tasked with inspecting Commitment Transactions associated with a user’s channel on the blockchain. Should an outdated Commitment Transaction be broadcasted, the watchtower possesses the authority to nullify it, acting as the user’s proxy, and subsequently penalize any deceitful actions. Nonetheless, this mechanism

is not without its constraints. The storage requirement for a watchtower entails the preservation of the most recent off-chain state for each channel, leading to linear memory usage and thereby posing scalability concerns. Moreover, potential for malfeasance arises if the watchtower either conspires with the user's counterparty or opts to minimize operational expenses. An unavailable watchtower can also inadvertently allow for misconduct.

Leinweber et al. (2019) suggest a framework for distributed watchtowers underpinned by a trusted execution environment (TEE). These TEE-supported distributed watchtowers leverage the low storage requirements of the TEE to improve the scalability of the watchtower. To guarantee the constant availability of the watchtower, the watchtowers interconnect to oversee one another, maintaining logs for user scrutiny. Nevertheless, Practical TEE deployments might have latent flaws, risking colluding for financial gain. Using game theory to attain a Nash equilibrium could address these concerns.

Liu, Szalachowski, and Sun (2020) introduce the concept of "Fail-safe watchtowers," engineered to thwart undesirable actions, even in the absence of the watchtower. This model introduces two temporal parameters: a shorter span, t (e.g., one hour), to ensure swift watchtower responses to channel termination requests and a protracted period, T (e.g., two days), which dictates the minimum time required for channel closure in the event of user indiscretions or watchtower failures. This, however, necessitates users to establish connectivity at least once within each T interval.

2.4.2 Transaction Fee Security

Malicious participants might not only misappropriate deposits from legitimate users within the channel but also divert transaction fees meant for honest participants. A study by Malavolta et al. (2019) introduces the "wormhole attack" which targets LN, enabling the attacker to siphon transaction fees from honest intermediaries within a designated payment trajectory. To clarify, let's assume a payment trajectory from sender to receiver as $(i_1, i_2, ..., i_n)$. Let's posit that i_m and i_k (with $k > m$) are two colluding intermediaries. The act of creating a HTLC with a subsequent node in the payment

path is termed the forward phase, while the act of disclosing the secret R and claiming funds from the preceding node is termed the backward phase. During the forward phase, each intermediary node creates an HTLC in alignment with the protocol, adjusting the forwarded amount value to allow for the payment fee. During the backward phase, the recipient and intermediaries sequentially disclose the secret R to their antecedents, halting only at the malicious intermediary i_k. Rather than divulging the secret R to the preceding intermediary (i_{k-1}), i_k confides it to i_m upon the expiration of i_{k-1}'s HTLC. Because of the HTLC's timeout, i_{k-1} presumes a payment failure, prompting all preceding nodes to fail the payment as well. But, because of i_k's revelation of the secret R, i_m doesn't fail the payment but proceeds to finalize the payment by disclosing the secret R to its antecedent. The nodes between k and m now miss out on the transaction fee and incur an opportunity cost for the time their funds have been locked up in the failed HTLC. The colluding nodes k and m gain the the transaction fees of the nodes in between. From the perspective of the sender and the receiver the payment succeeded as if nothing happened.

The Anonymous multi-hop lock (AMHL) (Malavolta et al. 2019), an innovative locking technique designed to segregate users within a payment trajectory, has been proposed as a deterrent to the wormhole attack (a comprehensive discussion is forthcoming in Section IV-B). Notably, the AMHL is compatible with LN. Maxwell et al. (2019) proposed the use of Schnorr signatures to replace the ECDSA scheme used in Bitcoin's original implementation. Using Schnorr signatures HTLC's can be made contingent on a public key (a *point* on Bitcoin's elliptic curve), hence the name PTLC: Point Time Locked Contract. Since PTLCs in a payment path are not all contingent on the same payment hash like HTLCs, using PTLCs would be a good countermeasure against the wormhole attack.

2.5 NETWORK ROBUSTNESS

If we look at LN we can describe its network as a weighted graph $G = (V, E)$ where V is the set of nodes in the network and E is the set of bidirectional payment channels. If we look at the basic properties of the LN graph we see a very low *transitivity* and *density* (Zhao, Zhou, and Su 2021). The *degree distribution* follows a power-law, from which we

can deduce that LN is a scale-free network. the fact that it is scale-free means two things, firstly the network is tolerant to random failures and secondly, the network is sensitive to targeted attacks (Albert, Jeong, and Barabási 2000).

Lee and Kim (2020) conducted an analysis on the LN's resilience to both random and specific adversarial actions, utilizing a game-based framework (Nagaraja 2007) with alternating stages of offense and defense. These stages, purely simulated, entailed node removal during the offense phase and their reintroduction during the defense phase. Specific adversarial strategies encompassed node targeting based on their degree, their betweenness centrality, and community structure (Fortunato and Castellano 2012; Girvan and Newman 2002; Porter, Onnela, and Mucha 2009). Countermeasures were developed considering nodes' degree and betweenness centrality. Simulations indicated that adversarial actions based on degree were most impactful, while defensive measures centered on centrality proved most effective. Notably, targeted strategies posed greater risks than random ones, consistent with the LN's scale-free characteristic.

The adversarial tactics outlined in Lee and Kim (2020), being simulations, didn't provide practical implementation methods. However, Mizrahi and Zohar (2020) introduces congestion-based adversarial tactics, which are potent given that a significant proportion of the LN's liquidity can be immobilized with less than 0.5 BTC.

Two main approaches exist for the congestion attack: one targeting the entire network by immobilizing numerous high-liquidity channels and another focusing on hub nodes by depleting the HTLCs in a targeted channel of the victim, effectively separating the victim's node from the graph. For broad network disruption, an attacker utilizes elongated payment paths, both endpoints controlled by the attacker, to dispatch numerous micropayments. The attacker then delays the final payment's execution, eventually revoking it. To target hub nodes, attackers repetitively route payments up and down through a node's target channel, consequently monopolizing its bandwidth for further payments.

An intuitive countermeasure against such congestion attacks is to decrease route lengths, especially since the average path between nodes in the LN rarely exceeds 10

hops. Minimizing the maximum permissible route length can both guarantee efficient routing and counteract congestion attacks. For comprehensive network attacks that exploit HTLCs in payment channels, one remedy is to prioritize trusted peers of nodes, enabling them to handle numerous simultaneous payments, while restricting unknown or less trusted nodes to processing fewer transactions. As for the hub node attacks, a promising solution entails preventing back-and-forth payment transfers within the same channel.

An alternative mitigation against this kind of griefing attacks that target the network robustness is explored in Mazumdar, Banerjee, and Ruj (2020). This paper suggests an alternative for the standard HTLC-contract called an HTLC Griefing-Penalty (HTLC-GP). An HTLC-GP contract enforces a collateral to be locked into the contract by the payer that is proportional to the collateral costs for the routing nodes of having an amount locked up in an HTLC. If the payee colludes with the payer and decides to grief by refusing to release the pre-image of the payment for a set period of time (e.g. one day), the HTLC-GP will pay out the griefing penalty to the last routing node. The last routing node in turn will have to pay out a smaller griefing penalty to the routing node before it, and so on. This results in having all routing nodes being fairly compensated for the incurred costs of having funds locked up. Nobody, except for the payer loses funds in order to compensate for the affected parties.

Shikhelman and Tikhomirov (2022) introduces another type of mitigation against this type of congestion attacks. Their solution combines unconditional fees and a measure of reputation, based on past behavior. Unconditional fees are small costs imposed on every payment attempt, regardless of whether the payment succeeds or fails. This strategy aims to deter quick jamming attacks by making it financially unfeasible to continuously send payments that block channels. Local reputation systems track the behavior of peers based on past interactions. This strategy addresses slow jamming attacks by punishing peers who consistently forward payments that take too long to resolve.

With unconditional fees make jamming costly, deterring attackers from continuously sending low-value jams to occupy slots or high-value jams to target liquidity; with the reputation system nodes are incentivized to forward payments promptly and honestly to

maintain a good reputation and avoid penalties.

This approach discourages congestion attacks while remaining compatible with the original incentive-based architecture of LN.

2.6 PRIVACY

Within LN, transactions modify the distribution of value in payment channels without recording them on the blockchain. At cursory inspection this seems to imply that the privacy in LN is improved compared to Bitcoin Layer-1. While true in principle, LN suffers from its own challenges with regards to privacy (Malavolta, Moreno-Sanchez, Kate, Maffei, and Ravi 2017). Based on the notions of interests with regards to Malavolta, Moreno-Sanchez, Kate, Maffei, and Ravi (2017)'s threat model (See also chapter 1.2.2), we define the privacy challenges of LN into three categories:

- Payment Value Privacy: Typically, intermediaries within the payment trajectory can ascertain the payment value, factoring in transaction fees; however, entities external to this trajectory remain unaware of the precise transaction sum.
- Relationship Anonymity: Utilizing the shared hash value of HTLCs in a payment path, intermediaries within a payment trajectory are able to deduce the respective payer and payee (Malavolta, Moreno-Sanchez, Kate, Maffei, and Ravi 2017; Green and Miers 2017). Furthermore, external entities can infer payment activities by observing fluctuations in the channel's liquidity metrics (Dandekar et al. 2011).
- Balance Privacy: Within the Lightning Network framework, the overall capacity of a payment channel is public, given the broadcasting of the Funding Transaction on the blockchain. Conversely, the internal allocation of funds within the channel remains private. Considering that a user's cumulative balance across all channels equates to her total assets in this network, preserving the confidentiality of individual channel balances becomes imperative. Yet, owing to the network's foundational architecture that mandates users to notify senders of successful or failed payments, a recipient retains the capability to discern the balance metrics of other users in respective payment channels (Herrera-Joancomartí et al. 2019)

(See also chapter a.).

2.6.1 Payment Value Privacy

As delineated in Malavolta, Moreno-Sanchez, Kate, Maffei, and Ravi (2017), in scenarios where only honest users partake in a payment, malicious entities external to the payment trajectory remain oblivious to the transacted value. However, a malicious intermediary embedded within this trajectory can easily discern the forwarded payment value. Consequently, to maintain the integrity of payment value privacy, it becomes paramount to ensure the honest participants along the entire payment pathway.

Roos et al. (2018) introduce 'SpeedyMurmurs', an innovative routing paradigm built upon prior studies (Malavolta, Moreno-Sanchez, Kate, and Maffei 2017; Roos, Beck, and Strufe 2016; Tsuchiya 1988), designed for decentralized payment channel networks like LN. This model aims to offer a subdued rendition of value privacy. Prior to transaction initiation, the originator identifies several pivotal nodes (those with extensive connectivity) within the system. Given L such nodes, the originator partitions the payment value into L segments, channeling each fraction through a trajectory over the respective landmark node. As such, adversaries, even if they compromise nodes within a subset of payment trajectories, can merely ascertain a minimum payment value rather than its exact amount. The partitioned transaction approach of SpeedyMurmurs increases the amount HTLCs used for payments over LN, negatively affecting the network throughput. This is an issue with all multi-part payment protocols and shows the always hard to find balance between privacy and utility.

2.6.2 Relationship Anonymity

The Lightning Network implements an onion routing protocol (Camenisch and Lysyanskaya 2006) (See also chapter 1.2.3), inspired by Sphinx (Danezis and Goldberg 2009). This protocol restricts intermediate nodes' knowledge to only their immediate preceding and succeeding nodes, preventing awareness of path length or their position therein.

However, adversaries can link senders to receivers through the unique payment hash value used along the payment route.

Both AMHL and PTLC as discussed in chapter 2.4.2, would be an effective countermeasure against this type of payment correlation. What these solutions have in common is a different release condition for every contract along the payment path, instead of the single release condition contingent on the payment hash for each contract along the payment path in the case of HTLC. Having different unrelated release conditions impedes the opportunity to use that information for payment correlation.

2.6.3 Balance Privacy

Within the Lightning Network framework, the payment channel's capacity is transparently available, because the funding transaction is broadcasted to the blockchain. The allocation of funds within said channel remains private exclusively to the channel partners. But the inherent properties of LN make it that it is possible to abuse the protocol in a way that allows for disclosing channel balances as explained below.

a. Balance Discovery Attack

The combination of unknown balances and source-based routing makes the process of finding a payment route one of trial and error. The payer needs to find a route to the payee, without knowing whether the individual balances along the route allow for relaying a payment with that specific amount. The Balance Discovery Attack (BDA) leverages this process of route finding.

When one of the relaying nodes has insufficient local balance to relay the payment it will return an `InsufficientFunds` or `TemporaryChannelFailure` error. If the balances *are* sufficient, the payment would succeed. This process of finding a successful route is called *probing*. But now consider a payment with a random *payment hash*. If all balances are sufficient, the payee would return an `UnknownPaymentHash` error since it is from an invoice it did not create. Receiving the latter error means that the

balances are sufficient for relaying the payment. Receiving either `InsufficientFunds` or `TemporaryChannelFailure` means they are insufficient. By repeating the payment with a binary search on the payment amount, we can identify the exact balance of a channel with a precision of a single millisatoshi.

In the basic scenario for channel balance discovery (Herrera-Joancomartí et al. 2019) it is assumed that there is an open payment channel AB between Alice, A, and Bob, B, with capacity C_{AB}. The goal of the adversary, Mallory, M, is to discover the balances of each node in channel AB: $balance_{AB}$ and $balance_{BA}$. To do so Mallory opens up a channel with Alice (see figure 2.3).

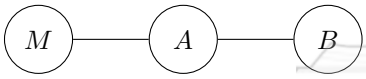

Figure 2.3 Basic BDA were the adversary Mallory tries to disclose the balance between Alice and Bob

Mallory tries to to disclose $balance_{AB}$ by routing invalid payments through $M \leftrightarrow A \leftrightarrow B$, using the basic BDA algorithm. The inputs for the algorithm are the target node B, the route to the target node, the value range to search in, given by 0 and C_{AB}, and the required accuracy for the algorithm. The algorithm creates invalid payments by using random, invalid payment hashes for each payment. The value for each payment follows a binary search pattern for which the initial lower and upper bounds are given by the value range input.

Bob, the recipient of the payment, is the only one who can determine that a payment from Mallory is invalid. Therefore, receiving an error stating the payment hash is invalid, means that $balance_{AB}$ was sufficient to route the payment, because if it was not, Alice would have returned an error stating insufficient funds and Bob would never have known about the payment. This fact is leveraged by updating the lower bound of the binary search to the value of the last payment. If however the failure message states insufficient funds, the upper bound is updated with the value of the last payment. This process repeats itself recursively until the difference between the upper bound and the lower bound of the binary search is within the threshold set by the accuracy input. The algorithm returns a tuple that gives the range within which $balance_{AB}$ sits.

Since the capacity of the channel C_{AB} is known, the $balance_{BA}$ can be calculated with $balance_{BA} = C_{BA} - balance_{AB}$.

Herrera-Joancomartí et al. (2019) showed that 89.10% of all channels could have their balances exactly disclosed with the application of the BDA. There appears to be room for increasing that level of disclosure by extending this attack.

By periodically executing a BDA, an adversary can monitor balances over time. This allows for tracing transactions. Therefore, this type of attack poses a threat for the value privacy as described in the threat model above.

b. Parallel channels & the generalized geometrical probing model

LN allows for peers to open up more than one channel between them. This could be warranted if a channel between two peers is depleted on one side. The channel partner on that side could open up a new channel, which would allow it to send payments again, while retaining the ability to receive payments through the other channel. This concept of parallel channels was not properly addressed in earlier BDA algorithms. Biryukov, Naumenko, and Tikhomirov (2022) tackled this and in doing so introduced the concept of the generalized geometric probing model.

The geometrical model is a model where all parallel channels in a N-channel hop are represented by an n-dimensional (hyper-)rectangle. This (hyper-)rectangle represents the result space of all possible balance vectors with one of its vertices at the origin and the other vertex at the coordinate represented by the exact balances of each of the parallel channels. A probe removes a (hyper-)square from the result space, either from the origin point or from the opposite vertex, depending on the probing direction. If the probe failed the result space is contained by the (hyper-)square (an upper bound) if it succeeded all possible points in the result space are outside the (hyper-)square (a lower bound).

In this model probing becomes a model of chipping away at the original n-dimensional (hyper-)rectangle until the smallest set of possible balance vectors remains. This does not necessarily mean that we get a set of cardinality 1. In general if $n > 1$ the

set can not be shrunk to a single point.

c. Countermeasures

One can discern that the successful implementation of the probing channel balance attack hinges significantly on the node providing explicit error feedback (specifically `UnknownPaymentHash` and `InsufficientFunds`) to the malicious party. An countermeasure would be to amalgamate the error notifications of `UnknownPaymentHash` and 'InsufficientFunds' into a singular, ambiguous message, withholding specifics about the channel where the fault transpired (Herrera-Joancomartí et al. 2019). Nonetheless, this approach presents a drawback: legitimate transacting parties, when faced with transaction failures, are left uninformed about the precise nature of the issue, potentially undermining the reliability of the payment routing system.

A second approach involves a node selectively declining a certain fraction of payment requests by following a drop rate parameter. By delivering a failure message devoid of diagnostic details, the cause of refusal remains ambiguous. This can deter an attacker, who may misconstrue the denial as a route failure due to insufficient funds.

This tactic essentially obfuscates data through the introduction of noise, rendering the attack uncertain or impractical. A balance exists between maintaining privacy and system functionality related to the drop rate. To optimize this, more discerning methods can be adopted in lieu of a random drop rate.

Indicators suggestive of a BDA, or similar threats, can guide these decisions. For instance, if node A notes an unusually high frequency of payment requests from node B, which has limited channels, it may view this as indicative of an attack and adjust its drop rate accordingly. Similarly, a suspicious sequence of payment amounts can also trigger enhanced drop rates (Herrera-Joancomartí et al. 2019). These represent basic heuristics to identify anomalous activities. More broadly, a node's actions can be characterized and anomaly detection metrics employed to adaptively modify the drop rate based on specific scenarios. Ultimately, node administrators determine the drop rate and the desired level of privacy, acknowledging the potential trade-off of occasionally declining legitimate

payments.

Thirdly, the fee system introduced by Shikhelman and Tikhomirov (2022) (See also: 2.5) could also be a mitigation strategy against BDA. Although mainly designed as a countermeasure for congestion attacks, both congestion attacks and BDA rely on sending many failed payments. Mitigations against either attack would probably translate to mitigating the other attack as well.

2.7 SUMMARY

This chapter gave a thorough review of the state of knowledge of LN technology and its challenges. We covered three main areas of interest: Transaction security, network robustness and privacy. We touched on the countermeasures that have been proposed in the academic literature.

We looked at the BDA specifically (Herrera-Joancomartí et al. 2019) and there appears to be room with this attack for an increased level of balance disclosure. this explains our research aim outlined in section 1.1 where we plan to replicate and extend the basic BDA. If we look at mitigations we see literature exploring mitigating congestion attacks Shikhelman and Tikhomirov (2022) (See also 2.5), but neither strategy is designed specifically with BDA in mind. This is why we defined our research aim specifically to target BDA.

According to this literature review, 89,10% of the channels are at risk of their balance being exposed (See section 2.6.3, a.). So when we look at the first stated research objective in section 1.4, our goal is to at least replicate that result and on to top of that find ways to disclose even a larger amount of balances. Looking at our second and third research objective, we can see that they relate to balance privacy (See section 2.6.3) in general and specifically to finding countermeasures (See section 2.6.3, c.) which is currently an unexplored avenue in the academic literature.

Later in this thesis (See 3.4 and 3.4.2) we will propose two countermeasures and show their effectiveness in preventing BDA. These countermeasures are two of the key

contributions of this thesis.

CHAPTER III

METHODOLOGY

3.1 OVERVIEW

This chapter lays out the methodologies used in this research. The goal is to gauge the impact of the BDA and to find feasible countermeasures. We start by exploring the LN protocol, the potential for extending the BDA and the tools for researching the role of the specific LN software clients in chapter 4.2. Then we lay a firm mathematical foundation under the first countermeasure that is of interest to us, namely differential privacy, in chapter 3.4. Finally we present our second countermeasure where we challenge the paradigm of the circuit switched network in LN and implement a packet switched alternative in chapter 3.4.2.

3.2 RESEARCH SETUP

An overview of the research setup is provided below (See figure 3.1) and subsequently explained in more detail.

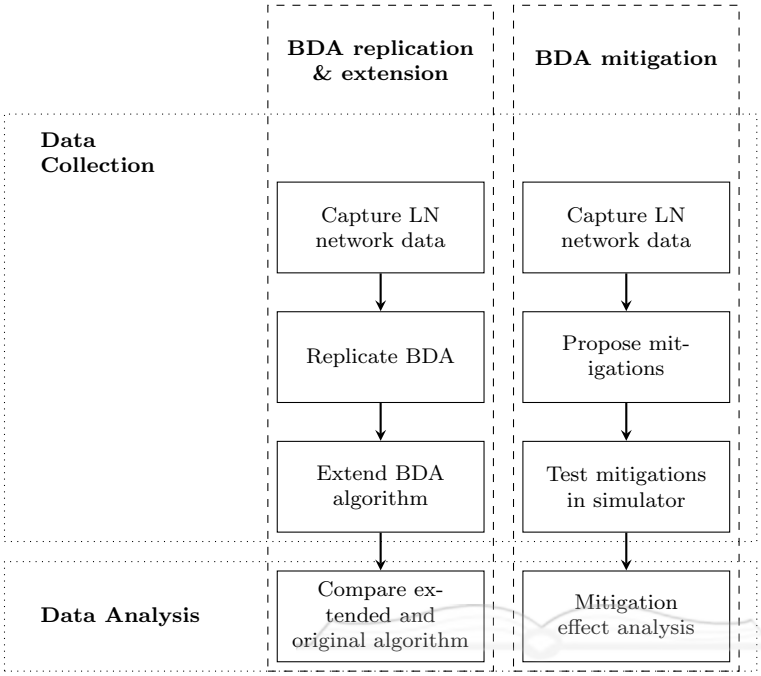

Figure 3.1 Schematic representation of the research methodology

3.3 EVALUATE THE EFFECTIVENESS OF BDA

In order to research the effectiveness of the balance discovery attack (BDA) in compromising off-path value privacy, we must first determine which LN clients are available. By determining the network share of each client type, we are able to determine the most important clients. We will then create a wrapper for each client to make it possible to control each client through their respective API's in a uniform way. Those clients are then used to run a local testing cluster of LN nodes. On the testing cluster we will run test scenario's to evaluate BDA and find novel ways to extend the algorithm.

On the testing cluster we will analyze the basic and extended algorithms having the LN nodes in each possible permutation of supported clients. This helps us determine if the new algorithm is to be considered an improvement and whether protocol implementation differences could play a role in BDA.

3.3.1 Establish network share of LN clients

In order to research the role of LN client software in BDA we must first determine which LN clients are available. We will use 1ML Lightning Network Search and Analysis Engine[ˆ1ml] to estimate respective proportions of each client in LN. 1ML is a website that publishes the current state of the LN graph and allows for node owners to self-report on a voluntary basis the type of client they use. We will choose the three top LN clients with the largest network share for analysis in our local test cluster. 1ML is a commonly used source for data in Layer-2 blockchain scaling research (Sguanci, Spatafora, and Vergani 2021; Zhao, Zhou, and Su 2021; Kumble, Epema, and Roos 2023; Ersoy, Moreno-Sanchez, and Roos 2023).

The 1ML data will be used as a sample of the population of LN nodes to determine the proportion of the different LN clients in the network. To determine the confidence interval we will apply a Correction Factor for Finite Populations (FPCF) because the population size is small (less than 1 million) and the sample size represents more than 5% of the population (Bondy and Zlot 1976). [ˆ1ml]: https://1ml.com/

3.3.2 Client Wrapper

The top three LN clients will be used to run certain test scenario's on. To run a scenario regardless of the type of software an LN node runs, we need a way to uniformly control each of the three clients.

All clients have an API that allows for communication with a local instance of a client. Although all clients follow the same LN specifications, the implementations of those API's differ on details. E.g. the commands used for connecting to peers, or opening up payment channels, and the parameters needed to do so, differ slightly across the three clients. Some commands run synchronously on one client while asynchronously on another. Therefore it is necessary to develop a wrapper for each of the clients' API, making it possible to address the API programmatically in a uniform way, independent of the type of client.

All three clients offer route discovery functionality, that searches the network graph for a feasible route for a given payment between two nodes. In a basic scenario, where three nodes are connected using two channels, it's a given which route will be returned, since there's only one possibility. It is to be expected that in more complex test clusters the discovered route might differ, depending on the implementation of the client. For a BDA to be successful, the route needs to be exactly as expected. We need to develop a route creation algorithm, that returns a route in the required format for each client along a predetermined route.

In the end the wrapper will support the commands listed in table 3.1.

Table 3.1 Different commands used by the top three clients, and their translation to a generic command used by the wrapper

generic	LND	Core Lightning	Eclair
getInfo	getinfo	getinfo	getinfo
connect	peers (POST)	connect	connect
disconnect	peers (DELETE)	disconnect	disconnect
listPeers	peers (GET)	listpeers	peers
openChannel	channels (POST)	fundchannel	open
closeChannel	channels (DELETE)	close	close
listChannels	channels (GET)	listchannels	channels
describeGraph	graph (GET)	listchannels [1]	allchannels & allnodes
newAddressIso	newaddress (GET)	newaddr	newaddress [2]
sendToRoute	channels/ transactions/ route (POST)	sendpay & waitsendpay	sendToRoute
getForcedRoute	-	-	-

Using the wrapper it is possible to implement several test scenarios, and run them on any possible configuration of different clients across multiple nodes.

[1]Core Lightning returns all channels out of the network graph as it is known to the client, when using listchannels, whereas the other clients return the channels of that specific node. The other clients have a separate command for returning the network graph as it is known to the client.

[2]Eclair doesn't support a wallet, instead it uses bitcoind's wallet

3.3.3 Testing cluster

The open source tool Simverse[3] will be used to create a testing environment for evaluating the BDA. Simverse generates a local cluster of LN nodes, each running one of three supported clients. Those clients align with the clients with the largest share. Each LN node runs in it's own Docker container and Docker-compose is used to manage the cluster.

The LN nodes use Bitcoin Core's Bitcoind implementation as a Bitcoin backend. Bitcoind will run in regression testing mode, known as regtest mode. This is a local test mode, making it possible to almost instantly create blocks with no real world value. Using regtest mode, the different implementations can be tested without incurring transaction fees for the on-chain transactions and without having to wait for blocks to be mined.

The testing cluster provides a realistic and flexible environment to rigorously test BDAs across different LN client implementations, ensuring accurate reproduction and extension of attack scenarios.

3.3.4 Replicate & extend the BDA

On the testing cluster, we will analyze both the basic and possibly new algorithms using all possible permutations of LN nodes with supported clients. This analysis will help us determine if the new algorithm constitutes an improvement and whether client differences influence the BDA. Replication of the Basic BDA is done two-fold, by doing measurements on a network snapshot and by implementing and executing the attack on our testing cluster.

To replicate the basic BDA, we conducted the same measurements on the same snapshot taken on January 8, 2019, derived from the original paper (Herrera-Joancomartí et al. 2019). The LN protocol impose two main limits: the maximum amount allowed for a single payment (MAX_PAYMENT_ALLOWED) and the maximum amount for creating a channel (MAX_FUNDING_ALLOWED), with values of 4,294,967 and 16,777,215 satoshis,

[3]https://github.com/darwin/simverse

respectively.

Considering these limits, we can assess whether a channel capacity in the snapshot exceeds this limit, and whether the basic BDA is not able to reveal the exact balance but rather provide a lower bound equivalent to MAX_PAYMENT_ALLOWED.

We will also prove the effectiveness of the basic BDA by implementing the original algorithm in code and executing it on our testing cluster.

Changes to the original algorithm will be analyzed from the point of view of the imposed limits and whether these still hold and validated against a new snapshot, subjected to the same measurements, taking into account any updated limits.

Changed algorithms will also be implemented in code and executed on our testing cluster to prove their effectiveness.

3.4 DEVELOP AND PROPOSE NOVEL MITIGATION STRATEGIES

In this study, we explore payment noising (by means of implementing approximate differential privacy) and payment splitting as two potential mitigations against BDAs in LN. Payment noising involves introducing controlled randomness into transaction amounts and frequencies, thereby obscuring the precise details of individual payments. Payment splitting, on the other hand, divides a single payment into multiple smaller transactions, making it more challenging for attackers to trace and aggregate balance information. Both techniques aim to enhance privacy by increasing the difficulty for attackers to accurately estimate channel balances. In the following sections, we will detail the methodologies used to apply these mitigations abd to evaluate the effectiveness of these mitigations in protecting against BDAs.

3.4.1 Approximate Differential Privacy

Differential privacy is a technique for the privacy-preserving, public disclosure of statistical information of a data set. Differential privacy ensures that this can be done without comprising the privacy of any individual by requiring that, whether any single person's record is or is not part of a database, the probability distribution of the published statistical information does not change in a meaningful way (Dwork 2006). This is usually achieved by adding noise to the disclosed output. By observing this noisy output, an observer is unable to discern if a specific record is in the database or not, regardless of the information this observer possesses about the other records in the database. This adding of noise is done through a probabilistic algorithm applied to the data set contained in the database.

Our goal is to obtain event-level, approximate differential privacy for the payments over a public LN payment channel. To achieve this, we will add random noise to each payment. The specific details of the implementation will be explained in detail later. For now, it suffices to say that we will noise the original payment with an extra payment that is atomic in the sense that an attacker cannot mount a BDA in between the original payment and the noise payment. The noise payment itself does not leave traces that an attacker can use to determine the amount of noise. This is achieved through either the use of private channels or through proposed changes to the LN protocol (chapter 5.1.9). It is an obvious requirement that the payee receives the original amount that was intended to be transferred. The noising mechanism should take this into account by making the noise transaction circular, meaning that the payer and payee are the same. So we have the original transaction that is relayed normally and a noise transaction that returns to the node creating the noise.

3.4.2 Payment Splitting

To enhance the chances of a successful transaction, the idea of splitting larger payments into smaller, more manageable portions was floated quite early in the development of LN. This was based on the rationale that several smaller transactions would have a

higher probability of success compared to a single large one. The benefit of increased payment success is enough on its own to pursue payment splitting, but it has also potential interesting side-effects on the BDA that we researched for this thesis.

a. Experimental Setup

The experimental setup was designed to evaluate the effectiveness of the payment splitting mechanism within LN. The setup involved the creation of a simulated LN environment where nodes could establish channels and perform transactions. The primary objective is to assess how payment splitting can mitigate BDA by leveraging the inherent properties of onion routing.

In this simulated LN environment all nodes support payment splitting. In a channel between Alice and Bob, both nodes indicated their support for payment splitting. This configuration allows for the subsequent splitting of payments into multiple smaller transactions, between Alice and Bob.

When Alice received a payment to relay, she decided to split the payment into two parts. One part follows the original route, while the other part (or parts) are sent to Bob via an alternative route. Alice forwards the original onion as usual but commits to a Hash Time-Locked Contract (HTLC) with an amount lower than the intended payment. Bob, upon receiving this partial payment, waits for an agreed-upon time instead of failing the HTLC, as he also supported payment splitting. Alice then sents the remaining amount to Bob using the same payment hash. Bob can only claim the payments once he learns the preimage used to create the payment hash. Upon receiving the second payment, Bob is economically incentivized to forward the first payment by setting up the next HTLC and forwarding the remaining onion of the first payment.

This methodology ensures that the PSS mechanism is thoroughly tested under various scenarios, allowing for a comprehensive evaluation of its impact on mitigating BDAs within the LN.

b. Lightning Network Probing Simulator

To evaluate the effectiveness of payment splitting we utilized the LN Probing Simulator. The LN Probing Simulator[4] was introduced by Biryukov, Naumenko, and Tikhomirov (2022), it uses a snapshot of the Lightning Network to recreate the actual graph at a specific time. Because the actual balances are hidden, they cannot be part of a snapshot, so the simulator samples a balance for each channel from a uniform distribution between 0 and the capacity of that channel. The simulator operates in two modes: direct probing and remote probing.

Direct probing means that the attacker opens a channel to one of the two nodes on either side of the channel being probed. Probes are then sent through a 2-hop path. Since all probes reach the target, direct probing is efficient, but it incurs on-chain fees for each channel being probed. It also requires the participation of the victim when opening a channel.

Remote probing means probing through a multi-hop path where the attacker isn't directly connected to one of the two nodes on either side of the channel being probed. By choosing wisely whom to connect to, an attacker can drastically reduce the amount of on-chain fees required to probe multiple target hops. It also allows for gathering data on the intermediary hops as well, which can aid in forming a network-encompassing view of balances Rahimpour and Khabbazian (2022). The drawback of remote probing is that there is a chance of probes not reaching their target, because of intermediary hops failing or lacking enough balance to relay the payment.

We forked the original simulator and created a version that integrated with the `LattE` software[5]. We reproduced the probing results of Biryukov, Naumenko, and Tikhomirov (2022) and compared them to probing PSS hops using the same LN snapshot. This is a snapshot created using a `Core Lightning` node on 2021-12-09. This snapshot contains 17068 nodes and 78076 channels which is in line with public network explores at the time.

[4]https://github.com/s-tikhomirov/ln-probing-simulator
[5]The repository is at https://github.com/gijswijs/ln-probing-simulator

By implementing payment splitting within this simulated environment, we were able to assess how splitting payments into smaller transactions and routing them through alternative paths impacted the BDA. This approach provided a controlled and reproducible setting to measure the performance and security benefits of payment splitting against traditional probing techniques.

We establish information gain for both probing methods (direct probing and remote probing) and in two contexts (PSS and non-PSS). For each combination, we simulate attacks on target hops with 1 to 5 parallel channels. From the network graph, we randomly select 20 target hops with the required amount of parallel channels and then simulate the probing. We repeat this 50 times and average the results. In total, we run $5 \times 20 \times 50 = 5000$ probes for each of the 4 combinations.

3.5 QUANTIFY THE PRIVACY IMPACT OF BDAS ON LN BY MEASURING PRIVACY LOSS

In this study, we employ three key measurements to evaluate the effectiveness of privacy-preserving techniques against Balance Disclosure Attacks (BDAs) in the Lightning Network: Differential Privacy, Information Gain, and Channel Information Coefficient (CIC). Differential Privacy provides a formal framework for quantifying the privacy guarantees of our methods by measuring the impact of individual transactions on the overall data distribution. Information Gain assesses the reduction in an attacker's uncertainty after a series of probes, thereby indicating the effectiveness of privacy measures in obscuring transaction details. The Channel Information Coefficient (CIC) quantifies the precision of an attacker's balance estimates, with lower values indicating greater privacy. Together, these metrics offer a comprehensive evaluation of how well our proposed mitigations protect against BDAs. The subsequent sections will delve into the specific methodologies used to calculate and interpret these quantities.

3.5.1 Differential Privacy On Streams

In the context of our study, we extend differential privacy to accommodate continual observation, where data is continuously produced and analyzed. This requires redefining adjacency for data streams, such that two streams are considered adjacent if they differ at exactly one time point (Chan, Shi, and Song 2011). This extension ensures that privacy is maintained even as new data is continuously incorporated into the system.

We quantify the privacy guarantee using the differential privacy parameter ϵ, with lower values indicating stronger privacy. By comparing the differentially private outputs with and without noising, we assess the extent to which noising payments enhances privacy. This methodology allows us to rigorously evaluate the trade-offs between privacy and data utility in a continually observed LN environment.

3.5.2 Information gain

We can measure the success of a probe by the amount of information gained through a successive series of probes. Information gain is calculated similar to Biryukov, Naumenko, and Tikhomirov (2022), by comparing the attacker's uncertainty U before and after the attack. The attacker's uncertainty is defined as the number of bits required to encode the cardinality of the possible result set. This is calculated by $log_2(f(P_i))$ where P_i is the polytope describing the attacker's current estimates after the i^{th} probe. After n probes, the uncertainty changes from $U_{before} = log_2(f(P_0))$ to $U_{after} = log_2(f(P_n))$. In case of a set of target hops T, we can sum uncertainties. The Information Gain then becomes:

$$I = 1 - \frac{\sum\limits_{t \in T} U^t_{after}}{\sum\limits_{t \in T} U^t_{before}}$$

By comparing the Information Gain with and without payment splitting, we can assess how effectively payment splitting reduces the attacker's ability to gain information, thereby enhancing the privacy of transactions.

3.5.3 Channel Information Coefficient

The Channel Information Coefficient (CIC) (Tikhomirov et al. 2020) is a measurement for the effectiveness of BDAs. It is defined as

$$i = 1 - \frac{b_{max} - b_{min}}{c}$$

where c is the channel's capacity and b_{max} and b_{min} are the upper and lower bound of the BDA estimates for the balance b. $i = 0$ would indicate no extra information on the balance other than what is announced through the gossip protocol. $i = 1$ would indicate perfect knowledge of the balance.

By introducing noise into payment amounts and frequencies, we aim to increase the range between b_{max} and b_{min}, thereby reducing the CIC value. This reduction in CIC demonstrates the effectiveness of noising payments in obscuring the true channel balances and enhancing privacy. Through this methodology, we can quantitatively assess how well noising payments protect against BDAs.

3.6 SUMMARY

This chapter presented the blueprint and approach adopted for the execution and delineated the architecture of the models involved. The investigational layout is grounded in the academic literature presented in the literature review. We have expanded upon the various steps that are involved in implementing the different extensions and countermeasures and elaborated on the expected results and the measures to identify and quantify those results.

Table 3.2 shows a summary of the experimental setup and the tools used for implementation and evaluation.

Table 3.2 Summary of experimental setup

Objective	Algorithm	Tools
Evaluate Effectiveness	BDA	Simverse

Objective	Algorithm	Tools
Evaluate Effectiveness	improved BDA	Simverse
Develop Mitigation	-	Python
Quantify Privacy Impact	-	LN Probing Simulator

CHAPTER IV

EVALUATION OF THE EFFECTIVENESS OF BDAS

4.1 OVERVIEW

The results of our evaluation reveal significant insights into the effectiveness of BDAs on the Lightning Network. Our findings highlight the extent to which attackers can infer channel balances and the conditions under which these attacks are most successful. The subsequent sections present a detailed analysis of our results, demonstrating both the strengths and limitations of existing privacy measures.

4.2 IMPROVED BDA ALGORITHM: TWO-WAY CHANNEL PROBING

The original algorithm (Herrera-Joancomartí et al. 2019) is bound by an upper limit set by MAX_PAYMENT_ALLOWED. This limit makes it impossible to probe balances that are higher than $2^{32} - 1\, msat$. This paper proposes an improved algorithm.

Consider a channel AB with capacity C_{AB}. Since $C_{AB} = C_{BA} = balance_{AB} + balance_{BA}$, the following holds

$$C_{AB} < 2^{33} \implies min\{balance_{AB}, balance_{BA}\} < 2^{32}$$

For all channels with a capacity $C_{AB} < 2^{33}$ there's always a balance lower than $\frac{2^{33}}{2} = 2^{32}$ on one end of the channel. With this knowledge we can extend the algorithm by letting it probe the channel from the other side, once we assess that the balance is higher than MAX_PAYMENT_ALLOWED on the initial probing side. This setup requires an optional second channel from the adversary Node M to Node B, to be able to probe the

channel between Node A and Node B from the side of Node B. (See figure 4.1)

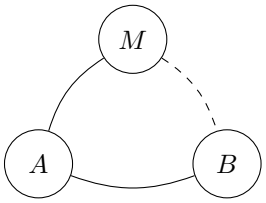

Figure 4.1 Basic scenario with an optional second channel for two-way probing

Algorithm 1 describes BDA with optional two-way probing for channels with a capacity above MAX_PAYMENT_ALLOWED. Algorithm 1 takes the same input parameters as the basic algorithm and returns the same tuple.z

If C_{AB} is higher than than MAX_PAYMENT_ALLOWED, the algorithm will try to send a fake payment with a size of exactly MAX_PAYMENT_ALLOWED. If that payment is possible, we have assessed that we are on the wrong end of the channel for probing the balance. The algorithm now calls itself with the target node and the final node of the route switched. The algorithm assumes that there is a route from the adversary to this new target node. The return value of that call is $balance_{BA}$, for calculating $balance_{AB}$ we use the following formula:

$$balance_{AB} = C_{BA} - balance_{BA}$$

If $C_{AB} >$ MAX_PAYMENT_ALLOWED and $C_{AB} < 2 \times$ MAX_PAYMENT_ALLOWED, the value of the first payment will not be exactly in the middle of the value range for the binary search, since it will use the fixed value of MAX_PAYMENT_ALLOWED for the first payment. That makes this algorithm slightly less computationally efficient then a perfect binary search, but it minimizes the use of the optional second channel.

4.3 RESULTS AND EVALUATION OF BDA REPLICATION AND EXTENSION

We confirmed the improvements provided by the two-way probing algorithm in two ways. Firstly we confirmed the feasibility of the algorithm in our local testing cluster. Secondly we analyzed LN running on top of Bitcoin mainnet, to estimate the number of channels

Algorithm 1 Two-way Probing

Data: route, target, maxFlow, minFlow, accuracy_treshold

Result: bwidth, an array of tuples that gives the range of bandwidth discovered for each channel

$missingTests \leftarrow True$
$bwidth.max \leftarrow maxFlow$
$bwidth.min \leftarrow minFlow$
$channelCapacity \leftarrow getInfo(target).capacity$
while missingTests **do**
  if $bwidth.max - bwidth.min \leq accuracy\_threshold$ **then**
    $missingTests \leftarrow False$
  end if
  if $bwidth.max \geq 2^{32}$ **then**
    $flow \leftarrow 2^{32} - 1$
  else
    $flow \leftarrow (bwidth.min + bwidth.max)/2$
  end if
  $h(x) \leftarrow RandomValue$
  $response \leftarrow sendFakePayment(route = [route, target], h(x), flow)$
  if $response = UnknownPaymentHash$ **then**
    if $bwidth.min < flow$ **then**
      $bwidth.min \leftarrow flow$
    end if
  else if $response = InsufficientFunds$ **then**
    if $bwidth.max > flow$ **then**
      $bwidth.max \leftarrow flow$
    end if
  end if
  if $bwidth.min = 2^{32} - 1$ **then**
    $newTarget \leftarrow route.pop()$
    $route \leftarrow route.push(target)$
    $bwidthBA \leftarrow twowayProbing(route, newTarget, bwidth.min, 0, accuracy\_treshold)$
    $bwidth.min \leftarrow channelCapacity - bwidthBA.max$
    $bwidth.max \leftarrow channelCapacity - bwidthBA.min$
    $missingTests \leftarrow False$
  end if
end while
return $bwidth$

that can have their balances disclosed by this algorithm and compare this to the earlier version of this attack.

4.3.1 Local Network Evaluation

We ran the Two-way Probing algorithm with every possible permutation of clients. By analyzing the responses from the clients, and analyzing the code of the respective clients on GitHub, we found that not every client implemented the MAX_PAYMENT_ALLOWED the same way.

On May 23rd, 2017 the BOLT specification was changed [1] by Paul "Rusty" Russel, who authored the majority of the BOLT documents. The variable containing the payment amount, amount_msat, was changed from a 32 bit unsigned integer to a 64 bit unsigned integer. This meant that before that change it was impossible to create a payment bigger than $2^{32} - 1$ whatsoever, but after that change in theory it was possible to create bigger payments. Additional specifications required the sending node to set the four most significant bytes of amount_msat to 0. But those additional requirements aren't implemented equally by the three main clients.

C-lightning is the only client that fully adheres to the requirements. Eclair has a limit of $5 \cdot 10^9 msat$. LND doesn't verify the amount for certain RPC's. By using the unverified RPC in our algorithm we could send fake payments up to the maximal channel capacity. This meant that we can disclose any balance between two LND Nodes, even if the balance is above the upper limit of the two-way probing algorithm. In the scenarios where Alice is a LND node and Bob is an Eclair node or both are Eclair nodes, balances up to $5 \cdot 10^9 msat$ can be disclosed without making use of two-way probing.

[1] https://github.com/lightningnetwork/lightning-rfc/commit/068b0bccf94e8cdaf5f298dade0fcc8cc8421ef6#diff-3369c5aa1774fef2ff1e246979f223eaR590

4.3.2 Channels affected

The two-way probing algorithm works regardless of the client software. So we can look at the channel capacity of all public channels in the LN graph and determine the proportion of channels that are now susceptible to this type of attack based on a snapshot of the network taken on the 3rd of October, 2019 (see figure 4.2).

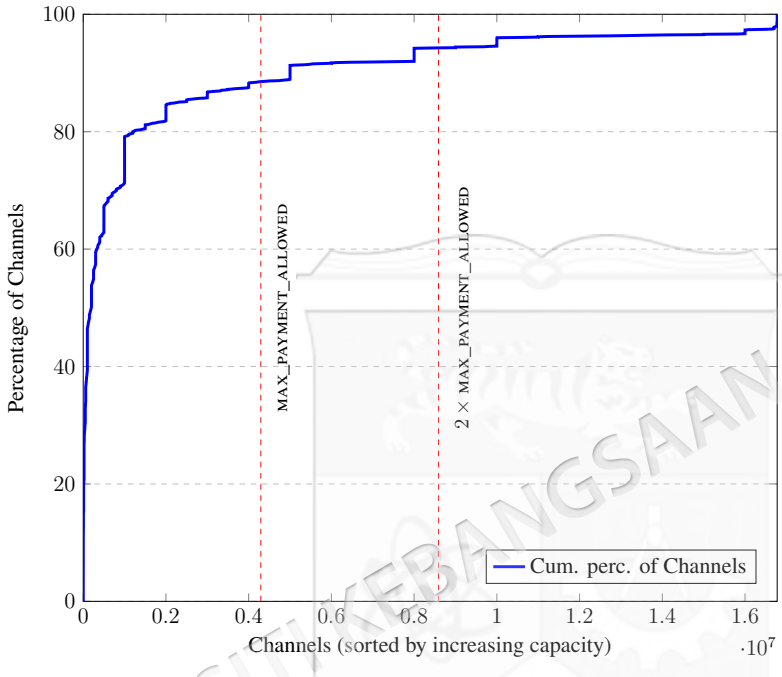

Figure 4.2 Cumulative percentage graph of payment channels ordered by increasing capacity. MAX_PAYMENT_ALLOWED shows the percentage of channels with disclosable balances using the basic BDA algorithm. $2 \times$ MAX_PAYMENT_ALLOWED shows the percentage of channels with disclosable balances using the two-way probing algorithm.

To estimate the number of channels susceptible for BDA above the 2^{33} limit set by the Two-way algorithm, we need to know the type of client on either side of a channel. There's no known way of figuring out what kind of client is installed, but if you know the proportion of each client type in the LN, it is possible to estimate the amount of channels for each specific combination of clients.

We queried 1ML for each node in our snapshot of the LN. We identified the client type for 273 nodes out of 3608 and estimated the proportion of nodes running different clients based on that data. (See table 4.1)

The LN is a graph G, with the number of vertices $n = |G(V)|$ and the number of edges $m = |G(E)|$. Our analysis yielded the following values for n and m:

$n = 3608$ $m = 9438$ with 1086 channels having a capacity greater than 2^{32} and 540 channels having a capacity greater than 2^{33}.

The client software defines the type of the vertex. $type_l$ for LND nodes, $type_c$ for c-lightning nodes and $type_e$ for Eclair nodes. An edge is said to be of $type_{(l,c)}$ if it connects a $type_l$ vertex and a $type_c$ vertex. The graph is without self-loops and undirected, so edge $type_{(l,c)} \equiv type_{(c,l)}$. Since we know the proportions of the different vertex types we can calculate the probability of an edge being of a specific type.

Table 4.1 Proportion of nodes running different Lightning clients

Client	n	Proportion (%)	CI[2] (%)
LND	220	80.59	(79.35-81.83)
c-lightning	40	14.65	(13.54-15.76)
Eclair	11	4.03	(3.41-4.65)
Other	2	0.73	(0.47-1.00)

- $P(type_{(l,l)}) = 0.8059^2$
- $P(type_{(c,c)}) = 0.1465^2$
- $P(type_{(e,e)}) = 0.0403^2$
- $P(type_{(l,c)}) = 2 \times 0.8059 \times 0.1465$
- $P(type_{(l,e)}) = 2 \times 0.8059 \times 0.0403$
- $P(type_{(c,e)}) = 2 \times 0.1465 \times 0.0403$

Assuming vertex type and channel capacity have a covariance of zero, the number of edges of each edge type, having a capacity greater than 2^{33} is calculated as follows: $P(type_{([c,e,l],[c,e,l])}) \times 540$. We are interested in the $type_{(l,l)}$ and $type_{(l,e)}$ channels, because the $type_{(l,c)}$ channels are susceptible to the Payment of Death explained below, which doesn't allow for discovering the balance. So the amount of channels with a capacity above 2^{32} is $540 \times P(type_{(l,l)}) + 540 \times P(type_{(l,e)}) = 386$ channels. So a total of

[2]95% Confidence interval

$9438 - 540 + 386 = 9284$ channels have balances that can be disclosed. This is 98.4% of all channels.

4.3.3 Payment of Death

In the case where Alice is a LND node and Bob is a C-lightning node, we saw interesting behavior of the C-lightning node which turned out to be a vulnerability of the current LN that can be exploited.

If a C-lightning node is being requested to route a payment to another node, or is the receiver of a payment, with an amount that his higher than MAX_PAYMENT_ALLOWED, it decides to fail the channel with the requesting node and close down that channel. Since LN uses onion routing, the requesting node from the perspective of the c-lightning node, is the one that comes just before it in the route. But that isn't necessarily the node from which the payment originated.

Consider the basic scenario (see figure 2.3), where Mallory and Alice run LND, and Bob runs c-lightning. Both channels between Mallory and Alice and between Alice and Bob have balances that allow for payments bigger than the MAX_PAYMENT_ALLOWED limit. If Mallory would create a fake payment with an amount above that limit, Bob would close down it's channel with Alice, without Alice being able to mitigate this in any way. We coined the term Payment of Death (POD) for this attack, after the infamous Ping of Death.

For the amount of channels affected by the POD we are interested in all $type_{(l,c)}$ channels, with a balance above MAX_PAYMENT_ALLOWED. This is $1086 \times P(type_{(l,c)}) = 256$ channels, meaning that 2.7% of all channels can be shutdown by using malformed payments.

We have notified the developers of the LN implementations by means of a responsible disclosure.

4.4 SUMMARY

Our evaluation of the effectiveness of BDAs on LN reveals the extent to which public channels are susceptible to these attacks. Utilizing a snapshot of the network from October 3, 2019, we analyzed the channel capacities and identified the proportion of channels vulnerable to BDAs. The two-way probing algorithm, which operates independently of client software, was applied to determine the channels at risk. By examining the client types and their distribution across the network, we estimated the number of channels susceptible to balance disclosure and the Payment of Death (POD) attack. Our findings indicate that a significant majority of channels, approximately 98.4%, are vulnerable to balance disclosure, while 2.7% of channels are at risk of being shut down through POD attacks.

CHAPTER V

BDA MITIGATION STRATEGIES

5.1 APPROXIMATE DIFFERENTIALLY PRIVATE PAYMENT CHANNELS

Differential privacy is a technique designed to enable the privacy-preserving public disclosure of statistical information from a dataset. It ensures that the inclusion or exclusion of any single individual's record does not significantly alter the probability distribution of the published data (Dwork 2006). This is typically achieved by adding noise to the output, making it difficult for observers to determine whether a specific record is part of the database, regardless of their knowledge about other records. Our objective is to implement event-level, approximate differential privacy for payments over LN channels. To this end, we will introduce random noise to each payment. The detailed implementation will be discussed later, but in essence, we will add an atomic noise payment that prevents attackers from executing a BDA between the original and noise payments. The noise payment leaves no detectable traces, achieved through private channels or proposed protocol changes. Importantly, the payee still receives the intended payment amount, as the noise transaction is circular, returning to the node that generated it.

5.1.1 Privacy-aware balance disclosure

First lets model our payment channel in a way that lets us reason about the privacy of its balance in the context of differential privacy.

A payment channel is created between two nodes in LN. When Alice opens a

channel with Bob, she creates a funding transaction. This funding transaction constitutes the opening balance of the channel. We assume that channels are single-funded. Both nodes can use the channel to perform payments, provided that there is enough balance on the side of the initiator of the payment. At any moment in time the balance reflects the current state of the channel.

To discern payments in LN, a possible adversary is interested in the balance of a channel in between transactions. A powerful enough adversary can detect the balance between each payment using a network-wide BDA. The adversary learns about each transaction executed in a channel by deducting the balance at time t from the balance at $t-1$. Given such a powerful adversary, we consider the transactions in our channel to be under continual observation of the adversary (Dwork et al. 2010).

We model our payment channel from the perspective of the funder. So transactions that add to Alice's balance are positive, while transactions that deduct from her balance (and add to Bob's balance) are negative, resulting in a stream of balances. We model this stream as $\sigma \in [1, n]^{\mathbb{N}}$. The value $\sigma(t)$ denotes the value of the funder's balance in millisatoshis at time $t \in \mathbb{N}$. Millisatoshi is the smallest unit of transaction in LN, so the

The transaction function is simply a function that returns the balance in a stream at point t.

Definition 1 (Transaction function)**.** *The deterministic transaction function* $c \to \mathbb{N}$ *returns the value of stream* σ *at point* t*:*

$$c(t) = \sigma(t).$$

5.1.2 Differential privacy on streams

In the original setting of differential privacy, the system is not expected to output results continuously. However, an LN node continually produces outputs — modeled by our transaction function — and is, as noted before, considered to be under continual observation. Therefore we need to achieve differential privacy under continual observation.

Differential privacy focuses on protecting any single individual; therefore, it relies on the notion of two databases being adjacent, i.e. differing only with regards to the information of one person. The set of all possible databases can be described by a *Hamming graph* and the distance $d_h(x, x')$ for any databases x and x' is exactly the number of persons in which they differ.

To extend our definition of differential privacy to include continual observation, we first need to extend the concept of adjacency to streams of data being produced whilst under continual observation. We say that two streams are adjacent if they differ at exactly one time t (Chan, Shi, and Song 2011).

Definition 2 (Adjacent Streams)**.** *The Hamming distance $d(\sigma, \sigma')$ is the number of elements in corresponding positions that differ in the two streams: $d(\sigma, \sigma') = |\{t : \sigma(t) \neq \sigma'(t), \forall t \in \mathbb{N}\}|$. Two streams σ and σ' are adjacent if and only if they differ at exactly one time t resulting in $d(\sigma, \sigma') = 1$.*

Having defined adjacency for streams we can now define differential privacy on streams, similar to Chan, Shi, and Song (2011).

Definition 3 ((ϵ, δ)-Differential Privacy on Streams)**.** *A probabilistic algorithm $\mathcal{M} \to \mathbb{R}$ provides (ϵ, δ)-differential privacy on streams if and only if for any two adjacent streams σ, σ' and $\forall S \subset \mathbb{R}$ we have*

$$Pr[\mathcal{M}(\sigma) \in S] \leq e^{\epsilon} \cdot Pr[\mathcal{M}(\sigma') \in S] + \delta,$$

where we denote with $\mathcal{M}(\sigma)$ the stream we get when applying $\mathcal{M}$ to each element of the stream σ individually. The probability is taken over the coin flips of $\mathcal{M}$

The Laplace distribution is the workhorse of differential privacy. We use it to add noise with the following mechanism. We will consider the implications imposed by the LN protocol on the noise mechanism later in this thesis.

Definition 4 (Laplacian Mechanism)**.** *Let $Lap(\frac{\Delta f}{\epsilon})$ denote the probability density function of the Laplace distribution with mean 0 and scale $\frac{\Delta f}{\epsilon}$. Then the function $\hat{c}(t) = c(t) + r$ with $r \leftarrow Lap(\frac{\Delta f}{\epsilon})$ is $(\epsilon, 0)$-differentially private.*

5.1.3 Modeling maximum payment size

The maximum payment one can make with LN is limited by the `amount_msat` variable. This is a 64-bit unsigned integer, with the four most significant bytes set to zero in the case of Bitcoin transactions. This leaves us with a maximum payment of $2^{32} - 1msat$. We use an additional LN transaction for putting into effect the Laplacian noise we add to our transactions, so the maximum payment size constitutes a limit to the amount of noise that can be added. Take note of the fact that the transactions to add noise are not transactions that can be discovered by a BDA. We will explain how to achieve this in chapter 5.1.9, for now it suffices to understand that in our model the noise transactions do not show up in our stream σ. The Laplacian noise generated by our Laplacian mechanism $\hat{c}(\sigma)$ can reach values bigger than this limit, which are impossible to accomplish with a single transaction. To overcome this issue, we define a function $\hat{c}_b$ that takes into account this limit.

Definition 5 (Maximum-payment-size-limited variant of $\hat{c}$)**.** *Given a function $\hat{c}$ with $\hat{c}(t) = c(t) + R$ for a deterministic function c and a random variable R. Given a limit for the maximum payment size b, we define the maximum-payment-size-limited variant $\hat{c}_b$ as follows:*

$$\hat{c}\_b = \begin{cases} \hat{c}(t) & \text{if } |R| \leq b \\ c(t) + b & \text{if } R > b \\ c(t) - b & \text{if } -R > b \end{cases}$$

5.1.4 Modeling channel capacity restrictions of the noise generating channel

Applying noise a single time is limited by the maximum payment size, but over time the total amount of noise added is also limited by the amount of money available for generating Laplacian noise. This amount is equal to the capacity of the channel that is used as a source for these noise transactions. If the channel is completely unbalanced, having the complete capacity on either side of the channel, it is obvious that it becomes impossible to either send or receive payments in that channel respectively. Hence, noise

that requires such a transaction would become impossible too. Continuing with $\hat{c}_b$ as in definition 5, we define an overall, bounded mechanism $\mathbb{C}$ that starts with an initial channel balance $bal(0)$ and adds noise as long as it does not exceed the capacity of the channel in either way. As soon as $\mathbb{C}$ does exceed the capacity $\mathbb{C}$ stops adding noise and outputs the bare transaction instead.

Definition 6 (Capacity Bounded Mechanism)**.** *: Given a function $\hat{c}$ with $\hat{c}(t) = c(t) + R$ for a deterministic function c and a random variable R, a capacity bound a and a maximum payment size b, we define the bounded mechanism $\mathbb{C}$ where $bal(t-1)$ is the balance before step t, R_t the noise added by $\hat{c}_b$ during step t and $s_k = \sum_{i=1}^{k} R_i$ the sum of all noise added until step k:*

$$\mathbb{C}(t) = \begin{cases} c\,(t) & \text{if } \exists k \leq t \text{ such that } s_k > a - bal(0) \vee -s_k > bal(0) \\ \hat{c}_b(t) & \text{otherwise.} \end{cases}$$

The new balance of the noise generating channel is $bal(t) \triangleq bal(t-1) + \Big(\mathbb{C}\Big(t\Big) - c\,(t)\Big)$.

The term *capacity bounded* should not be confused with *capacity bounded differential privacy* (Chaudhuri, Imola, and Machanavajjhala 2019). Throughout this thesis, the term *capacity bounded* denotes the bounds that arise from the capacity a of a payment channel and not the relaxation of differential privacy considered in the aforementioned paper.

5.1.5 Privacy analysis of maximum-payment-size obeying mechanism

In our maximum-payment-size obeying mechanism $\hat{c}_b$ the Laplacian noise is bounded. This will lead to the leakage of privacy. For instance, from large, noisy transaction values the adversary can deduce that the true transaction value can not be very small. The same holds the other way around. This leakage of privacy can be quantified using the concept of *statistical distance* (Backes and Meiser 2014).

Definition 7 (Statistical Distance)**.** *: The statistical distance between two random*

variables Y and Z is defined as

$$\Delta(Y, Z) \triangleq \max\_S \left| Pr\left[Y \in S\right] - Pr\left[Z \in S\right] \right|$$

Using the relationship between statistical distance and (ϵ, δ)-differential privacy, we can show that a function is differentially private based on that same property of another function.

Lemma 1 ((Backes & Meiser 2014)). *Given two probabilistic functions F and G with the same input domain, where F is (ϵ, δ_1)-differentially private. If for all possible inputs x we have that the statistical distance on the output distributions of F and G is: $\Delta\Big(F(x), G(x)\Big) \leq \delta_2$, then G is $\Big(\epsilon, \delta_1 + (e^{\epsilon} + 1)\delta_2\Big)$-differentially private.*

If we apply this knowledge to our unbounded function $\hat{c}$, we can define the (ϵ, δ)-differential privacy of $\hat{c}_b$ by finding the statistical distance between those two functions.

Lemma 2 ((Backes & Meiser 2014)). *Given an (ϵ, δ)-differentially private function $\hat{c}$ with $\hat{c}(t) = c(t) + R$ for a random variable R. Then for all t, the statistical distance between $\hat{c}$ and $\hat{c}_b$ is at most $\Delta\Big(\hat{c}, \hat{c}_b(t)\Big) \leq Pr[|R| > b]$.*

The statistical distance between $\hat{c}$ and $\hat{c}_b$ can be quantified, by using the properties of the Laplace distribution.

Lemma 3 ((Backes & Meiser 2014)). *Given a function with $\hat{c}$ with $\hat{c}(t) = c(t) + Lap(\frac{\Delta f}{\epsilon})$, the probability that the Laplacian noise $Lap(\frac{\Delta f}{\epsilon})$ applied to $c(t)$ is larger than the maximum payment size b is bounded by:*

$$Pr\left(Lap\left(\frac{\Delta f}{\epsilon}\right) > b\right) \leq \frac{2\left(\Delta f\right)^2}{b^2\epsilon^2}.$$

The proofs of all lemmas and theorems are postponed to appendix A.

5.1.6 Privacy analysis of capacity bounded mechanism

When the balance in the channel used as a source for the noise transactions — the noise generating channel — is insufficient, there is a breach of privacy. We can bound the probability for this unwanted event and integrate this in our overall results.

Bounding this event requires us to consider multiple steps over time. The probability of exceeding the capacity of the noise generating channel at time t depends on the combined noise being generated at all steps $< t$. Given infinite time any finite capacity will eventually be exceeded. We exclude the infinite case by only considering streams of transactions of a certain length n.

Again we use the concept of statistical distance and apply lemma 1 to determine the effect of the capacity bounded mechanism on differential privacy. We determine the statistical distance between $\hat{c}_b$ and $\mathbb{C}$. These functions differ on transaction streams of length n if and only if the capacity of the noise generating channel is exceeded at least once.

We start of by assuming that the noise generating channel starts with an optimal balance at $bal(0) = \frac{a}{2}$. The probability of exceeding the capacity of the noise generating channel is bounded by the probability that the sum of noise being added in all steps exceeds $bal(0) = \frac{a}{2}$.

Lemma 4. *Given a $(\epsilon, 0)$-differentially private function $\hat{c}$ with $\hat{c}(x) = c(x) + Lap(\frac{\Delta f}{\epsilon})$. If the corresponding bounded mechanism $\mathbb{C}$ has capacity bound a and a maximum payment size b, then for all streams σ of length n, the statistical distance between $\mathbb{C}$ and $\hat{c}_b$ when starting with a channel balance $bal(0) = \frac{a}{2}$ is at most*

$$\Delta\Big(\mathbb{C}\left(\sigma\right), \hat{c}_b\left(\sigma\right)\Big) \leq Pr\left[\exists k \leq n \left|\sum_{j=1}^{k} \hat{c}_b\left(j\right) - c\left(j\right)\right| > \frac{a}{2}\right].$$

Given a stream of transactions of a certain length n it is not sufficient to give a bound on the probability to exceed the capacity of the noise generating channel in one of the steps. We want a bound on the probability that the the capacity is exceeded at least

once over the complete length of the stream: We have to consider all steps at once.

Lemma 5. *Given a* $(\epsilon, 0)$*-differentially private function* $\hat{c}$ *with* $\hat{c}(x) = \sigma(x) + Lap(\frac{\Delta f}{\epsilon})$. *For all* $t > 0$*, the probability that the Laplacian noise exceeds the capacity* a *for* $a \geq 2(n+t) \cdot \frac{\Delta f}{\epsilon}$ *in at least one of the* n *steps is bounded by*

$$Pr\left[\exists k \leq n \left| \sum_{j=1}^{k} \hat{c}_b(j) - c(j) \right| > \frac{a}{2}\right] \leq \frac{2n}{t^2}.$$

5.1.7 Overall privacy guarantee

We can combine our analyses of the maximum-payment-size limited mechanism and the capacity bounded mechanism to obtain an overall privacy guarantee. We consider transaction streams of length n and require the noise generating channel to be optimally balanced at $bal(0) = \frac{a}{2}$ at the beginning. theorem 1 then follows directly from the analyses.

Theorem 1. *Given a* $(\epsilon, 0)$*-differentially private function* $\hat{c}$*. If the corresponding bounded mechanism* $\mathbb{C}$ *has capacity bound* a *and maximum payment size* b*, then* $\mathbb{C}$ *is* (ϵ, δ)*-differentially private on all streams of length* n *with* $\delta = (e^{\epsilon} + 1) \cdot (P_b + P_c)$ *where* P_b *is the statistical distance between* $\hat{c}$ *and* $\hat{c}_b$ *and* P_c *is the statistical distance between* $\hat{c}_b$ *and* $\mathbb{C}$.

Using lemma 2 and lemma 3 we obtain a value for P_b. Similarly lemma 4 and lemma 5 yield a value for P_c. With these values we achieve concrete bounds on differential privacy.

5.1.8 Capacity restrictions of the primary channel

The primary channel — the channel being noised — obviously has a capacity as well, and its restrictions should also be taken into account. If the balance of the primary channel is disproportionately one-sided, it could be unable to absorb the noise transaction.

This would result in a privacy breach that is similar to the breach of privacy caused by insufficient balance in the noise generating channel. Consider the situation where the primary channel is between Alice and Bob with capacity a' and with capacity a for the noise generating channel. Both capacities are equal ($a' = a$) and the balance of both channels is $bal_{ab}(0) = \frac{a}{2}$. In this state, any transaction would leave the balance in the primary channel in a state where it *possibly* is not able to absorb the noise. So we should consider this channel as being depleted from either side as if it had a capacity of zero.

But if we pad the primary channel with additional capacity equal to the capacity a of the noise generating channel we can make sure that the noise can be absorped.

Theorem 2. *Given a non-noised channel c with capacity b and a noised channel c' with capacity $b + a$ and a noise generating channel with capacity a , then any sequence of transactions of length n that is approved in c is approved including the additional noise in c'.*

We assume that the primary channel satisfies these conditions. We can then disregard the possibility of a primary channel that can not absorb the noise generated by the noise generating channel.

5.1.9 Implementation

For a noise mechanism to work, it needs to be able to firstly fulfill the noise payment without the adversary being able to mount a BDA in between the actual payment and the noise payment. Secondly, after a noise payment has succeeded, the adversary should not be able to determine the amount of the noise payment.

We will sketch three approaches that achieve those goals.

a. Circular payments

In this approach, the operator of a differentially private channel is required to operate two nodes. A public node and node that we will call a private node, although that is not a

notion that is supported by the LN protocol. The private node has a private channel with the public node. The public node is used for opening public channels with other nodes. The private node simultaneously opens a private channel with the same node. This way the public channel has two private channels associated with it that offer an alternative route between the same two nodes with the private node as an intermediary hop. Consider Alice who runs node A and A' that are connected through a private channel. Alice wants to open up a channel with Bob, who runs node B. Alice does so by opening up a public channel between A and B and a private channel between A' and B. The noise payments can now be constructed as circular payments that start and finish in A, either by the route $A \rightarrow B \rightarrow A' \rightarrow A$ or the route $A \rightarrow A` \rightarrow B \rightarrow A$. It is a matter of convention to associated one of the routes to positive noise payments and the other to negative noise payments. We will use the convention that negative noise results in a noise payment that goes in the opposite direction of the actual payment (figure 5.1).

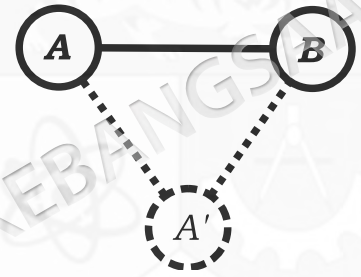

Figure 5.1 Circular noise payment through an extra node.

We have implemented this approach in a prototype using a c-lightning plugin. The pugin is represented by a conceptual model in pseudocode algorithm 2. This prototype is compatible with LN today.

b. Parallel channels

This approach is only possible with LN clients that support parallel channels. Of the three main clients, LND and Eclair support parallel channels, whereas c-lightning does not (Biryukov, Naumenko, and Tikhomirov 2022). In this setup there is no need to operate a second node. If Alice opens up a public channel with Bob, Alice opens a second private channel with Bob as well. The noise payment can now be constructed as a payment that goes up one channel and down the other.

Algorithm 2 Differential Private Channel Plugin

Data: $\Delta f, \epsilon, maxNoisePayment, privateNodeId$

```
1:  ourNodeId ← GET_NODE_ID
2:  procedure PAYMENT_SUCCESS(payment) ▷ An event listner for successful payments
3:      noisePayment ← isNoisePayment(payment)
4:      if not noisePayment then
5:          createNoisePayment(payment)
6:      end if
7:  end procedure

8:  procedure ON_HTLC_ACCEPTED(payment)        ▷ An event listner for HTLC relays
9:      noisePayment ← isNoisePayment(payment)
10:     channelOpen ← isChannelOpen(payment)
11:     if not noisePayment && not channelOpen then
12:         Throw error
13:     else
14:         Continue HTLC
15:     end if
16: end procedure

17: procedure createNoisePayment(payment)          ▷ Create noise payment
18:     noiseAmount ← Laplace(Δf/ϵ)
19:     if noiseAmount < 0 then
20:         noiseAmount ← min(abs(noiseAmount), maxNoisePayment)
21:         route ← getRoute((ourNodeId, payment.peer.Id, privateNodeId, ourNodeId))
22:     else if noiseAmount > 0 then
23:         noiseAmount ← min(noiseAmount, maxNoisePayment)
24:         route ← getRoute((ourNodeId, privateNodeId, payment.peer.Id, ourNodeId))
25:     end if
26:     SendPayment(route, noiseAmount)
27: end procedure
```

We could further optimize the noise payment if multipath payments (Wang et al. 2019; Di Stasi et al. 2018) are supported. Specifically Link Multipath Payments, as these are geared towards parallel channels. With Link Multipath Payments a node will relay a payment when the incoming HTLC's over the parallel channels *combined* are sufficient. Using this concept we can forego the negative noise payment by having a Link Multipath Payment that consists of an HTLC over the public channel for the amount of the actual payment minus the noise and an HTLC over the private channel for the amount of the noise (figure 5.2).

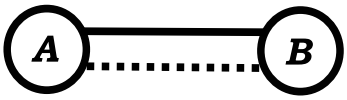

Figure 5.2 Noise payment through parallel channels.

c. Protocol change

A third approach would be a change to the LN protocol allowing to earmark part of the capacity of a channel for noise generation. Now instead of a single balance a channel would keep state of two balances. One is the main balance as we now know it, but with a capacity of the funding transaction minus the capacity for noise generation. The other balance is the noise generation balance. A payment through this channel would consist of two HTLC's one updating the main balance with the amount adjusted with the noise, and the other HTLC would update the noise generating balance. This approach is similar to the parallel channels approach but internalized in the LN protocol. With differential privacy being part of the protocol, peers can agree on opening a differential private payment channel as part of the other channel parameters peers have to agree on when opening a channel. Part of this agreement could be to have these channels be dual-funded by definition so that a channel is differentially private from the get-go. These kinds of optimizations are out of scope for this research.

5.1.10 Evaluation of Approximate Differential Privacy

To evaluate approximate differential privacy we first look at the feasibility of setting up noise channels, by graphing the minimal channel capacity of the noise channel as a function of the amount of privacy ϵ we want to achieve, under a fixed sensitivity. This allows us to quantify the cost of these channels. Ultimately we also look at how utility is affected by approximate differential privacy.

a. Minimal channel capacity

The capacity a of the noise channel is given by $a \geq 2(n+t) \cdot \frac{\Delta f}{\epsilon}$ and the capacity of the primary channel is at least $2a$. If we fix the sensitivity at the amount that we want to hide,

we can plot the minimal channel capacity as a function of ϵ. Using this information we plotted the minimal channel capacity for the noise channel and the primary channel with sensitivity at $2000sat$, $4000sat$ and $6000sat$ in figure 5.3.

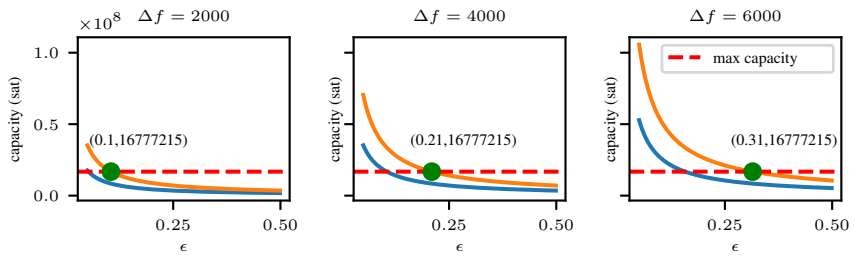

Figure 5.3 Minimal channel capacity with different Δf, and $n = 20$ and $t = 200$.

Figure 5.4 shows the relation between ϵ and δ on one side and the channel capacity of the noise channel on the other side.

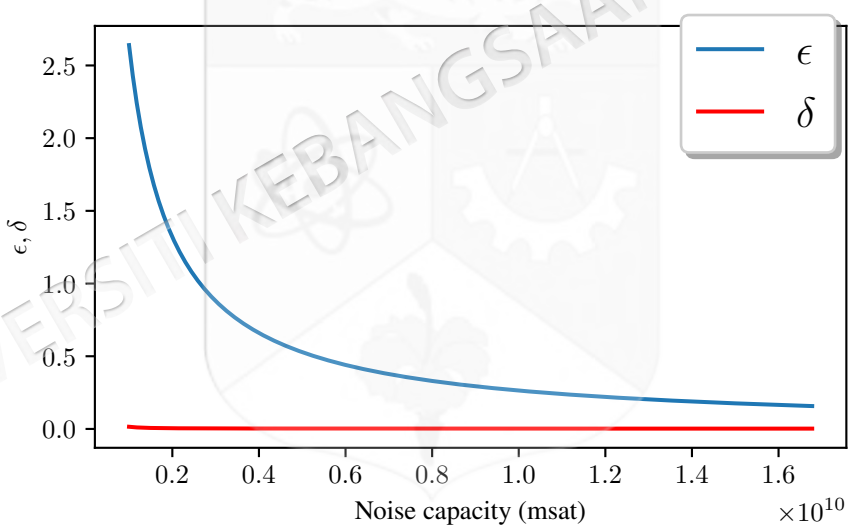

Figure 5.4 Privacy-efficiency trade-off

b. Cost

To quantify the cost of differentially private channels, we consider a normal channel with no noising mechanism to a noised channel. The capital cost of running a differentially private channel consists of three components. Firstly there are on-chain fees associated with opening and closing the private channels that will operate as a source for noise.

Secondly, the noise transactions incur a payment fee. Thirdly the capital locked up loses value at the inflation rate (figure 5.5).

For the on-chain fees, we only consider the funding and closing transaction of the channel. We assume that the channel closing is cooperative, so we disregard the possibility of a commitment transaction with HTLC success and timeout transactions needed to be broadcast to the blockchain. We also assume that the closing transaction has two outputs. The size of a funding transaction depends on inputs being spent and this is dependent on the funder. For simplicity, we assume a funding transaction that funds a single channel (we disregard the possibility of funding multiple channels with a single funding transaction) and spends a single input. Given these assumptions, we can make reasonable estimates for the size as measured in *weight units* (WU). We use a weight of $610WU$ for the funding transaction and a closing transaction weight of $722WU$. Getting these transactions mined in a reasonable time requires a transaction fee. The fee/WU can be very volatile depending on the congestion of the mempool. For our calculation we assume a fee/WU of $1sat$.

The LN fee incurred by the noise transactions is relative to the amount of the transactions, which are random. But we can work with the expected value of the sum of the noise transactions. The μ of the Laplacian noise is zero, but we incur fees regardless of the direction of the noise transaction, so we want to know the expected value of the absolute value of the noise. Taking the absolute value gives the following probability density function:

$$|Lap(s, x)| = \lambda e^{-x\lambda}$$

Given $\lambda = \frac{\epsilon}{\Delta f}$ which has an expected value of $\frac{\Delta f}{\epsilon}$, the expected value of the sum is

$$E\left[\sum^{n}\left|Lap\left(\frac{\Delta f}{\epsilon}\right)\right|\right] = n \cdot \frac{\Delta f}{\epsilon}.$$

But since ϵ is dependent on the capacity a of the noise channel.

$$\epsilon = \frac{2f(n+t)}{a}$$

We get

$$
\begin{aligned}
E\left[\sum^{n}\left|Lap\left(\frac{\Delta f}{\epsilon}\right)\right|\right] &= n \cdot \frac{\Delta f}{\epsilon} \\
&= n \cdot \frac{\Delta f \cdot a}{2\Delta f(n+t)} \\
&= n \cdot \frac{a}{2(n+t)}
\end{aligned}
$$

The final cost component is the decline in the time value of money. We assume an interest rate of 3 percent compounded annually. For a capacity of a, one has to pad the actual payment channel with $2 * a$ extra capacity to absorb the noise. The actual noise channel is a single channel in the case of parallel channels (chapter b.), or two channels in the case of circular payments (chapter a.).

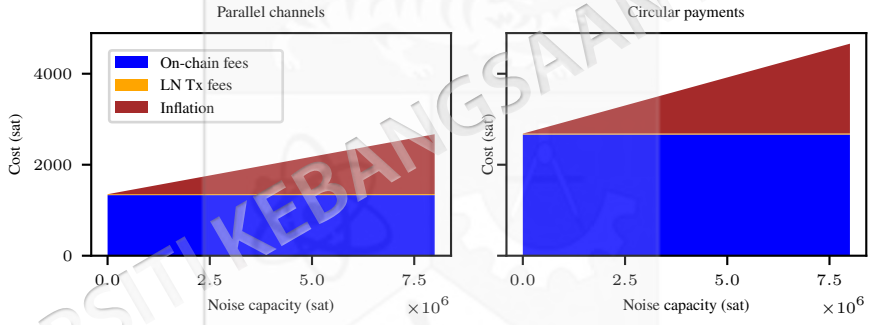

Figure 5.5 Expected capital cost per day.

c. Utility

A common way of measuring the performance of PCN's is by obtaining the *success rate*, the fraction of transactions it successfully processes (Malavolta, Moreno-Sanchez, Kate, and Maffei 2017; Roos et al. 2018; Sivaraman et al. 2018). If we assume that a payment always succeeds after a route with enough credit has been found, the success rate is only dependent on a route with enough credit being available and the path finding algorithm being able to find it. Obviously, our solution does not affect the path finding algorithm, so our solution can only affect the success rate through changing the balances in such a way that there is not enough credit available to route a payment.

d. Channel Information Coefficient

(Tikhomirov et al. 2020) defined the Channel Information Coefficient (CIC) as a measurement for the effectiveness of BDA's. It is defined as

$$i = 1 - \frac{b_{max} - b_{min}}{c}$$

where c is the channel's capacity and b_{max} and b_{min} are the upper and lower bound of the BDA estimates for the balance b. $i = 0$ would indicate no extra information on the balance other than what is announced through the gossip protocol. $i = 1$ would indicate perfect knowledge of the balance. Our solutions does not prevent a BDA, so a strong enough adversary would always be able to obtain a CIC of 1. But the noised balance, although fully disclosable, does not yield the same information as an unnoised balance.

Therefore, we calculate the the *maximal* obtainable CIC as follows:

$$i = 1 - \frac{|b_{noised} - b_{unnoised}|}{c}$$

.

For the difference between the noised balance and the unnoised balance, we can again use the expected value of the absolute value of the noise. This gives

$$i = 1 - \frac{\Delta f}{c\epsilon}$$

for the maximal obtainable CIC. We can now plot the maximal obtainable CIC as a function of the capacity and ϵ for different sensitivities (figure 5.6).

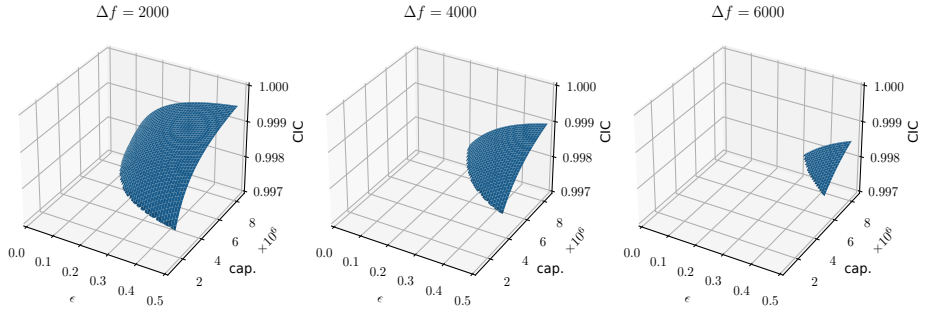

Figure 5.6 Channel Information Coefficient.

5.2 PAYMENT SPLITTING AND SWITCHING

The onion routing employed by LN prevents nodes in a payment path from knowing anything more than its predecessor and its successor in the payment path. It also prevents an intermediary node to change the payment path to use an alternative route to the destination, because there is no way for an intermediary node to know the destination of the payment. In that sense, LN acts as a circuit switched network.

It is that property that is leveraged in a BDA: The Sender chooses a payment path and does that so that a single hop can be targeted for probing.

We suggest a way for intermediary hops to split a payment into multiple smaller payments and send those smaller payments over alternative routes to the next hop. This is called payment splitting & switching (PSS). The workings are explained below.

5.2.1 Packet Switching Payments

Upon opening a channel both nodes signal that they support PSS. Let's assume that Alice and Bob have opened such a channel. Now, upon receiving a payment that needs to be relayed, Alice decides she wants to split the payment into two separate parts. By definition, one part will have to follow the original route as intended and the other part will be sent to Bob via an alternative route. Alice will start by forwarding the original onion as normal, but she will commit to an HTLC that carries an amount lower than the intended amount. Bob, upon receiving the amount, sees that the amount is insufficient to cover the amount that he needs to forward and the fees that he expects to receive for forwarding, but instead of failing the HTLC, he waits for an agreed-upon time, because Bob also supports PSS. Alice now sends a new payment to Bob for the remaining amount, using the same payment hash. Bob is the recipient of this payment, so Alice can create the entire onion, but Bob can only claim the payments once he learns of the preimage used to create the payment hash. Once Bob receives this second payment he is economically incentivized to forward the first payment. He does so by setting up the next HTLC and forwarding the remaining onion of the first payment, which is the original onion

containing the intended recipient of the payment.

The above method creates a localized packet switching of sorts where a node can use any route available to the next node, to forward a payment in separate parts. We have developed a plugin that shows the feasibility of this countermeasure. The plugin is represented by a conceptual model in pseudocode algorithm 3.

5.2.2 Probing of PSS hops

Assuming an attacker knows it is probing a PSS channel from Alice to Bob, it needs to change its conceptual model of what it is probing. Instead of knowing the route of a payment, it now needs to take into account that a payment can go through each and any of the possible routes from Alice to Bob at once. If Alice and Bob have other, parallel channels (Biryukov, Naumenko, and Tikhomirov 2022) then those qualify as alternative routes, but even routes via intermediary hops qualify as alternative routes. So instead of probing a single channel, an attacker is now probing the total liquidity of Alice in the direction of Bob.

If we take the probing model for parallel channels (Biryukov, Naumenko, and Tikhomirov 2022) as a starting point, we can think of each possible route as a single channel in a hop. Originally a hop referred to all the channels shared by a pair of adjacent nodes, but now we take all possible routes from two adjacent nodes into account. For clarity, we will refer to such a hop as a routing-hop.

The capacity of a route is defined by the channel with the lowest capacity of all channels in that route. The balance of a route is defined by the channel with the lowest balance in the direction of the route. This need not be the same channel. A routing-hop with n routes is defined by its route capacities $C = [c_1, ..., c_n]$ and balances $B = [b_1, ..., b_n]$. Balances are assumed to be in the direction of the node with the alphanumerically smaller ID. We define this direction as $dir0$. $dir1$ is defined as the opposite direction.

This definition collapses with the original definition for hops with parallel

Algorithm 3 Payment Splitting & Switching plugin

```
splitPayments ← []                                   ▷ Array of split payments to receive
procedure ON_HTLC_ACCEPTED(payment)                  ▷ An event listner for HTLC relays
    if payment.hashinsplitPayments then
        COLLECTPARTS(payment)
    end if
    peerSupportPSS ← payment.peer.features.pss
    if not peerSupportPSS then
        return                                       ▷ Handle payment normally
    end if
    altRoutes ← FINDALTROUTES(payment.peer.Id)
    splitAmounts ← SPLITAMOUNTS(payment.amount, len(altRoutes) + 1)
    payment.amount ← splitAmounts[0]
    SENDMSG(payment.peer.id,' SPLIT_PAYMENT_ANNOUNCE', payment.hash)
    for i ← 1, LEN(altRoutes) − 1 do
        SENDPAYMENT(altRoutes[i], splitAmounts[i], payment.hash)
    end for
    Continue HTLC                                    ▷ Continue original payment with adjusted amount
end procedure

procedure ON_CUSTOMMSG(peerId, msgType, payload)     ▷ An event listner for
received messages
    if msgType =' SPLIT_PAYMENT_ANNOUNCE' then
        splitPayments[payload.hash] ← peerId
    end if
end procedure

procedure COLLECTPARTS(payment)
    amt ← payment.amount
    fwdamt ← payment.forwardAmount
    rec ← splitPayments[payment.hash].get('totalReceived', 0)
    splitPayments[payment.hash].totalReceived = rec + amt
    if amt < fwdamt then                             ▷ This is the original payment
        splitPayments[payment.hash].totalExpected ← fwdamt
    end if
    if splitPayments[payment.hash].totalReceived ≥
splitPayments[payment.hash].totalExpected then
        Forward original HTLC as normal
    end if
end procedure
```

channels (Biryukov, Naumenko, and Tikhomirov 2022), in the case where the set of parallel channels equals the set of routes between two adjacent nodes.

There is a noteworthy difference between channels in a hop and routes in a routing-hop: If a channel i is enabled in both directions, the forwarding ability of a hop in $dir0$ is determined by its balance b_i, and its forwarding ability in $dir1$ is determined by $c_i - b_i$. In a *route* that is enabled in both directions, however, that relationship between capacity in $dir0$ on one side and balance and capacity in $dir1$ on the other side doesn't necessarily hold. The balance in $dir0$ and $dir1$ can be anything from 0 to the route capacity c_i and there need not be any relationship between the two. To fix this we model a bidirectional route as two unidirectional routes, for which we calculate the balance such that $b_i^{dir1} = c_i - b_i$ holds, but is meaningless in either $dir0$ or $dir1$, since the route is not enabled in that direction.

E^d is defined as the set of routes enabled in direction d, where $d \in \{dir0, dir1\}$. The forwarding ability of a routing-hop is set by the sum of balances of all routes enabled in a given direction, where h stands for the forwarding ability in $dir0$ and g in $dir1$.

$$h = \sum i \in E^{dir0} b_i$$
$$g = \sum i \in E^{dir1} (c_i - b_i)$$

Probing reveals information about h or g, and in some cases about individual balances. The attacker maintains the current lower and upper bound for all of them: $h^l < h \leq h^u$, $g^l < g \leq h^u$ and $b_i^l < b_i \leq b_i^u$. Before probing those bounds are set to

$$h^l = g^l = -1$$
$$\forall i \in \{1, \ldots n\}.\ b_i^l = -1$$
$$\forall i \in \{1, \ldots, n\}.\ b_i^u = c_i$$
$$h^u = \sum i \in E^{dir0} c_i$$
$$g^u = \sum i \in E^{dir1} c_i$$

For routes that consist of just one bidirectional channel the relationship between balance (in $dir0$) and forwarding ability in $dir1$ *is* meaningful. We can think of this forwarding ability as the balance in $dir1$ and this balance also has upper and lower bounds that the attacker can maintain when probing from $dir1$. These upper and lower bounds of the balance in $dir1$ relate to the lower and upper bounds of the balance in $dir0$ respectively:

$$b_i^l = c_i - b_i^{u_{dir1}} - 1$$
$$b_i^u = c_i - b_i^{l_{dir1}} - 1$$

It should be pointed out that all lower bounds are strict and all upper bounds are non-strict. Let F be the set of all possible values for B, considering the knowledge of the attacker. F is then the n-fold Cartesian product defined by $F = X_1 \times \ldots \times X_n = \Big\{(x_1, \ldots, x_n) | x_i \in (b_i^l, b_i^u] \text{ for every } i \in \{1, \ldots, n\}\Big\}$.

Assuming all available routes are used to their potential for a probe leads to the following updates of those bounds depending on whether the probe was successful or a failure:

a. Successful probe of amount a in $dir0$

A successful probe in $dir0$ has a direct impact on the lower bound h^l, since at least the probed amount has to be available for forwarding payments in $dir0$. Lower bounds being strict, the bound is adjusted to the amount probed minus one.

A successful probe potentially gives an attacker knowledge about the lower bounds of the balances of the possible routes. The intuition is that if we look at a single route out of all routes enabled in $dir0$, and the combined potential for all *other* routes enabled in $dir0$ (the sum of their b^u's) is not enough to relay the probed amount, then the remaining amount of the probe would have to be relayed through this single route. In that case, we

can update the lower bound of the balance for that single route. We can do this for every single route in the set of routes enabled in $dir0$.

A successful probe can also be used to update the upper bound g^u. But in this case, we have to look at whether all routes enabled in $dir0$ are also enabled in $dir1$ ($E^{dir0} \subseteq E^{dir1}$). If this is the case, then we can state that the total amount of the probe is not available for forwarding payments from "the other side", so we can update the upper limit g^u by deducting the probed amount from the total capacity of all routes enabled in $dir1$. If some of the routes enabled in $dir0$ are not enabled in $dir1$ we cannot make such a strong statement, so we resort to using the sum of upper bounds of forwarding capability in $dir1$ of all routes enabled in $dir1$, using the relationship between this upper bound and the lower bound of the balance in $dir0$, as explained above.

$$h^l = a - 1$$

$$\forall i \in E^{dir0}.\ b_i^l = \max\left(a - \sum_{\{j \in E^{dir0}: j \neq i\}} b\_j^u, 0\right) - 1$$

$$g^u = \begin{cases} -a + \sum\limits_{i \in E^{dir1}} c_i & \text{if } E^{dir0} \subseteq E^{dir1} \\ \sum\limits_{i \in E^{dir1}} c_i - b_i^l - 1 & \text{otherwise} \end{cases}$$

b. Failed probe of amount a in $dir0$

A failed probe in $dir0$ has a direct impact on the upper bound h^u since the probed amount a cannot be available for forwarding payments in $dir0$. The upper bound is adjusted to the amount probed minus one.

Likewise, a failed probe can potentially be a reason to adjust the upper bounds of the balances of the possible routes. Consider the case where we look at a single route of all routes enabled in $dir0$, and the total amount of which we are certain that it is available for all the *other* routes enabled in $dir0$ (the sum of their b^l's) is not enough to relay the probed amount, then the remaining amount of the probe would have to be relayed through

this single route and would have failed. So we can update the upper bound of the balance for that single route. We can do so for every single route in the set of routes enabled in $dir0$.

A failed probe can also be used to update the lower bound g^l. If we look at all routes enabled in $dir0$ that are bidirectional ($E^{dir0} \cap E^{dir1}$), we know for sure that the combined capacity of those bidirectional routes minus the probed amount increased by one, has to be on the "other side", otherwise the probe would have succeeded.

$$
\begin{aligned}
h^u &= a - 1 \\
\forall i \in E^{dir0}.\, b_i^u &= \min\left(a - 1 - \sum_{\{j \in E^{dir0} : j \neq i\}} (b_j^l + 1), c_i\right) \\
g^l &= \max(-a + \sum_{i \in E^{dir0} \cap E^{dir1}} c_i, -1)
\end{aligned}
$$

c. Successful probe of amount a in $dir1$

A probe from the other direction would lead to similar, but mirrored adjustments to the upper and lower bounds, but since we keep the balance of a route and its upper and lower bounds by definition in $dir0$ we need to adjust for this fact.

A probe in $dir1$ would lead to the following results:

$$
\begin{aligned}
g^l &= a - 1 \\
\forall i \in E^{dir1}.\, b_i^u &= \min\left(c_i - a + \sum_{\{j \in E^{dir1} : j \neq i\}} c_j - b_j^l - 1, c_i\right) \\
h^u &= \begin{cases} -a + \sum\limits_{i \in E^{dir0}} c_i & \text{if } E^{dir1} \subseteq E^{dir0} \\ \sum\limits_{i \in E^{dir0}} b_i^u & \text{otherwise} \end{cases}
\end{aligned}
$$

d. Failed probe of amount a in $dir1$

Finally, a failed probe in $dir1$ would lead to the following adjustments:

$$g^u = a - 1$$
$$\forall i \in E^{dir1}.\ b_i^l = \max\left(c_i - a + \sum_{\{j \in E^{dir1} : j \neq i\}} (c_j - b_j^u), -1\right)$$
$$h^l = \max\left(-a + \sum_{i \in E^{dir0} \cap E^{dir1}} c_i, -1\right)$$

e. Geometrical model of PSS probing

We can capture the probing of PSS hops in a geometrical model. An n-route hop can be seen as an n-dimensional (hyper-)rectangle R, where the sides run parallel to the axes. Each side corresponds to one possible route. Along the i^{th} dimension, R is defined by the point $[0, c_i]$. The origin is one of the vertices of R and the point $[c_1, \ldots, c_n]$ is the vertex diagonally opposite of the origin. Each lattice point inside R corresponds to a possible balance vector. The size of the possible result set is defined by the cardinality of the set of lattice points inside R.

A probe of amount a cuts the hyperrectangle along a hyperplane orthogonal to the all-ones vector $\vec{1}$. If the probe fails all points from the opposite vertex (in *dir0*) or the origin (in *dir1*) to the hyperplane, including all points in the hyperplane itself are removed from the result set. A failed probe results in a new upper bound, and since upper bounds are non-strict, the points in the hyperplane itself should be removed from the result set. If the probe succeeds all points on the opposing side of the hyperplane, are removed from the result set. Any adjustments in the bounds for the individual balances of the routes in the hop are captured by changing the dimensions of R. Specifically for dimension i, R will be defined by $\left[b_i^l + 1, b_i^u\right]$.

Figure 5.7 shows a 2-dimensional case with $c_1 > c2$ and a failed probe in *dir0* of amount a. The dashed diagonal line represents the hyperplane running through all

coordinates in the first quadrant of the Cartesian plane (including $[0, a]$ and $[a, 0]$) for which the sum of its coordinates equals a. Where the line intersects with R, the lattice points on the line correspond to a possible balance vector v for which $\sum_{i=1}^{n} v_i = a$. Since it is a failed probe, the upper right corner of the rectangle R is removed from the possible result set, leaving the colored polygon representing the attacker's current best estimates.

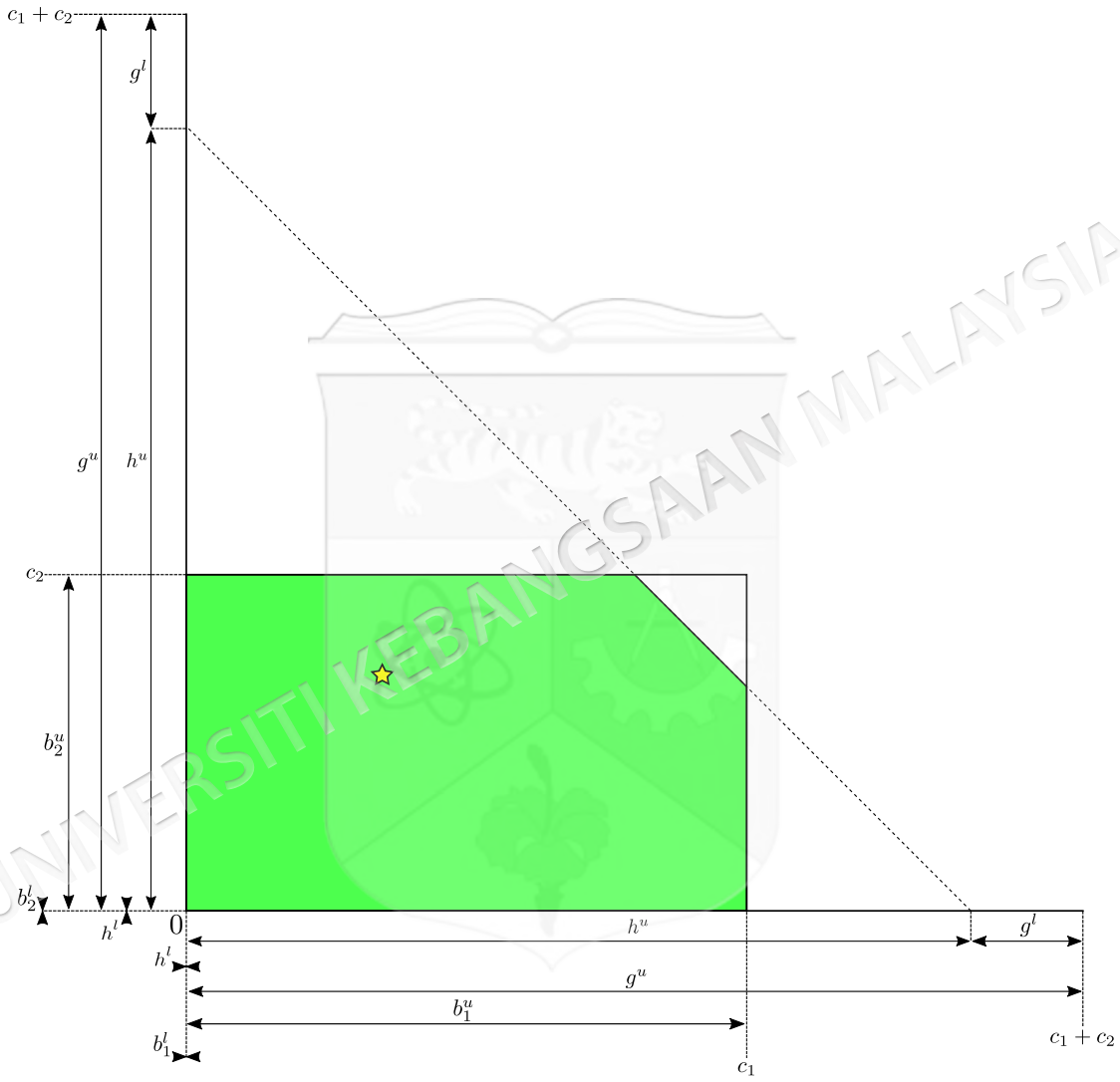

Figure 5.7 Failed probe in a 2-dimensional rectangle, resulting in a new upper bound.

Figure 5.8 shows a new probe in *dir0*, but this time it is successful. this results in the lower right corner of the rectangle R being removed, again leaving the colored polygon representing the attacker's current estimates.

Finally in figure 5.9 we see a state of probing where the current h^l and h^u

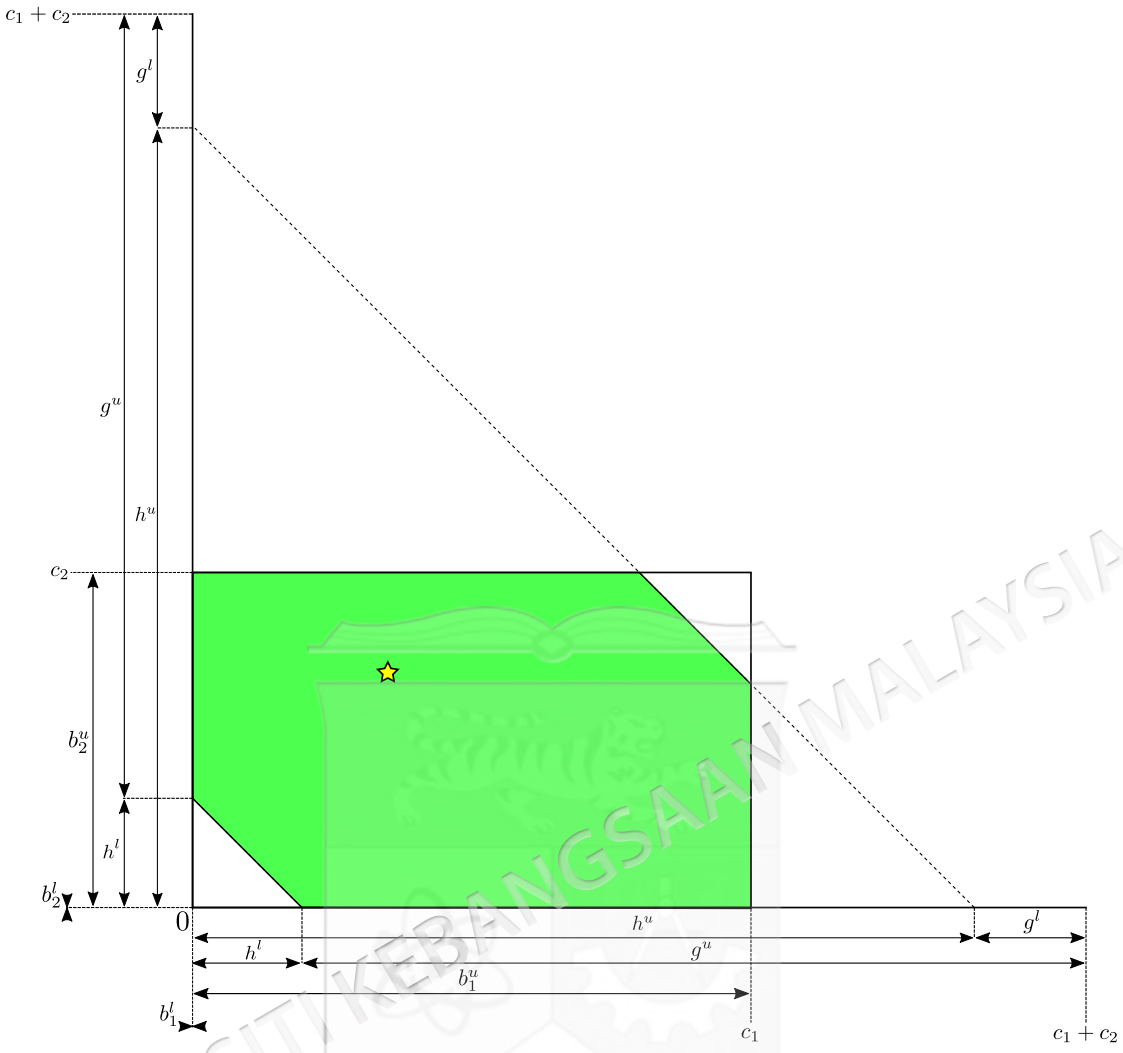

Figure 5.8 Successful probe in a 2-dimensional rectangle, resulting in a new lower bound.

necessitate an update to b_i^l and b_i^u

f. Convex polytope described through h-representation

We can describe the polygon (or polytope when the number of dimensions is larger than 2) representing the attacker's current estimates as a system of inequalities. Let P be that polytope. Its system of inequalities is described by $Ax \leq b$ where $A \in \mathbb{Z}^{m \times d}$, $A = (a_{ij})$, and $b \in \mathbb{Z}^m$. Assume that we are probing a routing-hop with two routes with capacity $c_1 = 160$ and $c_2 = 100$, similar to the example used in Figure 5.7, 5.8 and 5.9. We are probing from *dir0* with an amount of 80. We can now describe the polytope P as follows.

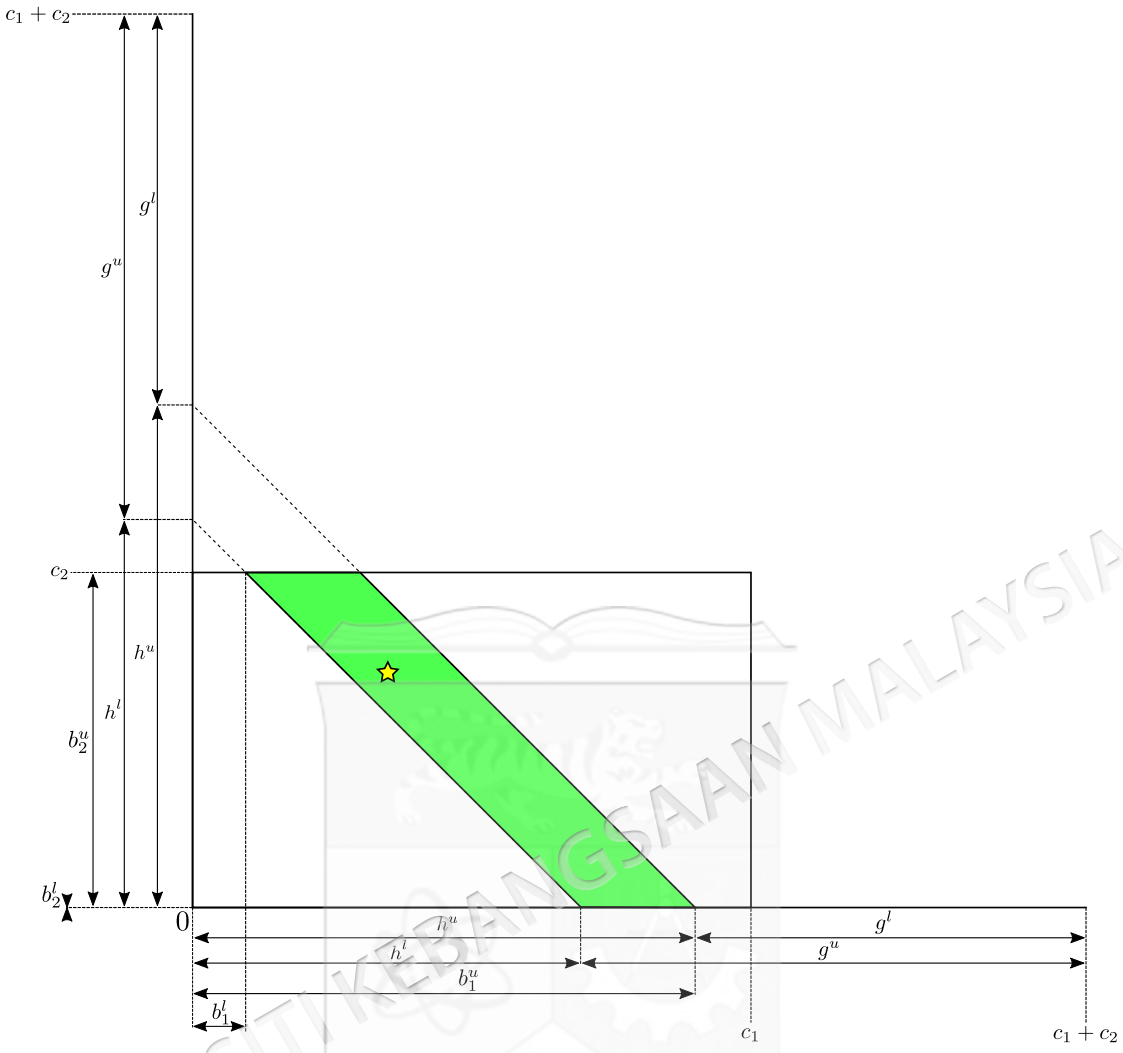

Figure 5.9 Probing state in a 2-dimensional rectangle, resulting in updated balance bounds for channel 1.

$P = \{(x, y) : x \leq 160, y \leq 100, x + y \leq 80, x \geq -1, y \geq -1\}$. Thus

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}, b = \begin{pmatrix} 160 \\ 100 \\ 80 \\ 0 \\ 0 \end{pmatrix}$$

Using Barvinok's algorithm (Barvinok 1994) we can find a polynomial time algorithm f for counting lattice points in a convex polytope for any polytope in n-dimensional Euclidean space $\mathbb{R}^n$. The computer package `LattE`, contains an implementation of A.

Barvinok's algorithm (De Loera et al. 2004). We use this package to determine the cardinality of the set of possible balance vectors given by $f(P)$.

5.2.3 Evaluation of Payment Splitting and Switching

PSS is evaluated using information gain. Information gain is calculated by comparing the attacker's uncertainty U before and after the attack. By comparing our results with the results from Biryukov, Naumenko, and Tikhomirov (2022), which is the current state-of-the-art in probing results, we can clearly quantify the effect of PSS as a countermeasure on the BDA.

5.2.4 Achieved Information Gain

When we look at the results (figure 5.10) we see that non-PSS probing, both direct and remote, are exactly in line with the results from Biryukov, Naumenko, and Tikhomirov (2022). This was expected, since we replicated the exact method of probing from that paper. To be able to compare our results with theirs, we have looked at the number of channels in the original target hops. So a target hop with one single channel in the non-PSS settings would be compared to a target hop with one single channel in the PSS setting. This does explicitly not mean that there is a single channel in the *routing-hop*. A routing-hop consists of *all* possible routes from two adjacent nodes. So what figure 5.10 shows us is the difference between probing a hop assuming no PSS, versus probing a similar hop assuming PSS.

The results are very pronounced for both direct and remote probing. With PSS, the information gain for direct probing of a single channel hop drops from 0.98 to 0.49 ($p < 0.0001$) and for remote probing it drops from 0.91 to 0.35 ($p < 0.0001$). When the number of parallel channels in the target hop increases the information gain decreases in non-PSS and PSS probing alike making the impact of PSS relatively smaller, but the difference remains distinct and significant. With PSS, direct probing a 5-channel hop drops from 0.27 to 0.22 ($p < 0.0001$) and remote probing drops from 0.16 to 0.13

($p < 0.0001$).

Moreover, the results for PSS-probing are upper bounds of the actual information gain. Since counting lattice points is computationally expensive, even when using Barvinok's algorithm (Barvinok 1994), we limited the total amount of routes in a routing-hop to a maximum of 9. Potential target hops with more routes in their routing-hop were disregarded. Higher dimensionality polytopes lead to lower information gain, hence the results of PSS-probing are an upper bound.

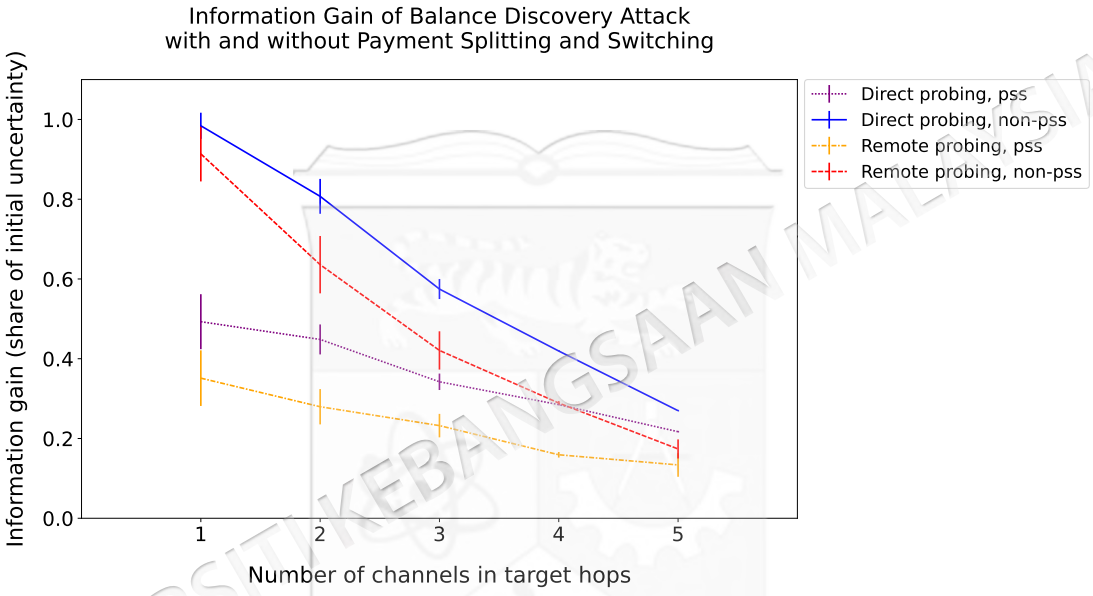

Figure 5.10 Information gain for direct and remote probing in PSS and non-PSS context

CHAPTER VI

DISCUSSION

6.1 OVERVIEW

This chapter interprets and describes the significance of our findings in relation to the current state of knowledge in academic literature. It will explain any new understanding and insights that resulted from this research and links them to the original research objectives stated in chapter 1.4.

6.2 REPLICATING AND EXTENDING THE BDA

Our first research objective was to replicate and extend the BDA. We extended the BDA by introducing two-way BDA, and we replicated the BDA from earlier work (Herrera-Joancomartí et al. 2019) by using it on a current snapshot of LN graph. This allows for comparing the two algorithms in relation to their effect on balance disclosure.

In our search for an interaction effect between the variable of which type of client software being used and the amount of channels that can have their balances disclosed, we not only found that there is indeed such an effect, explaining part of the increase of disclosure, but that there was also a completely new vulnerability that was previously unknown.

Finally, countermeasure suggested in earlier work (Herrera-Joancomartí et al. 2019) are infeasible or at the least troublesome in relation to the cost of operating and utility of the network.

6.2.1 The effect of two-way BDA on balance disclosure

Herrera-Joancomartí et al. (2019) reported that 89.10% of all channels could have their balances exactly disclosed. Our research showed that we can increase this to 98.37% of all channels, a 9.27 percentage point rise (See table 6.1). The basic BDA performed slightly less in our snapshot of the LN network because in the period between the two snapshots of the 8th of January, 2019 and the 3rd of October, 2019, the percentage of channels with a capacity C of $C > 2^{32}$ slightly increased.

Table 6.1 Percentage of channels susceptible for the basic BDA and the two-way probing BDA

Disclosable channels	basic BDA (%)	two-way probing BDA (%)
$C \leq 2^{32}$	89.10	88.49
$C > 2^{32} \wedge C \leq 2^{33}$	0	5.79
$C \leq 2^{33}$	0	4.09
TOTAL	89.10	98.37

6.2.2 Impact of Payment of Death

The properties of this specific vulnerability make it so that the highly capitalized nodes are more vulnerable, since it are these nodes that have channels with a balance above MAX_PAYMENT_ALLOWED limit. The average capacity of those 1086 channels is 10196116 $msat$. Using that average combined with the estimated proportions of affected channels, 17.5% of the total capacity of the network could be taken down with an organized attack. These proportions align with proportions found earlier through alternative methods (Pérez-Solà et al. 2020). It's reasonable to assume that these channels are responsible for routing a disproportionate amount of the payments on the network. Such an attack could have substantial impact on the ability to route payments of the network as a whole.

The closing of channels comes at a cost to the victim nodes, since you have to

broadcast on-chain transactions for closing a channel and again for reopening it. Those transactions have transaction fees attached to it. Furthermore, channel age is used as heuristic for determining the reliability of a node for routing payments, so routing nodes have an incentive to keep channels open as long as possible.

Clients should adhere to the BOLT specification, making it impossible to create payments with a value higher than MAX_PAYMENT_ALLOWED and deny to consider payments with a value above the MAX_PAYMENT_ALLOWED for routing. The latter would make it impossible to disclose balances above $2^{33}\, msat$. Secondly, clients should not consider a malformed payment a reason for permanently closing down a channel. This would make it impossible to mount a POD attack. It is with a sense of pride that we can share that at the time of writing of this thesis all three main clients have released patches to fix this particular type of vulnerability. This is one of the practical implications that our research has already had beyond the academic setting.

6.2.3 Suggested countermeasures from earlier work

Limiting the number of payment requests per unit of time or randomly deny payment requests following a configurable *dropping rate* (Herrera-Joancomartí et al. 2019) are unfeasible. These countermeasures, implemented generically, come at a high cost for usability of the network as a whole. So a selective approach is preferable, identifying nodes that are the source of suspicious payment requests, and limit or deny requests from these nodes. But this selective approach could quite easily be circumvented by inserting different non-malicious nodes with known balances in the route in subsequent payment requests. This way the source of the attack from the point of view of the attacked node is not determinable and each non-malicious node itself doesn't see a pattern that resembles an attack.

6.3 POSSIBLE COUNTERMEASURES

Our second research objective was to propose feasible and effective countermeasures against the BDA. We explored two such countermeasure, namely approximate differential privacy and payment splitting and switching.

6.3.1 Approximate Differential Privacy

Our research shows that it is possible to hide payments of reasonable size using differential privacy techniques. Noising techniques such as the ones described in this thesis could be used to achieve that goal. In this section we discuss the utility of our solution through the lens of the *truthful probability* of the channel balance, a term coined by (Tang et al. 2020). We look at the limitations of our solution and we look at the impact of our solution on the effectiveness of BDA's.

a. Utility

(Tang et al. 2020) uses a different utility metric based on the notion of the *truthful probability* of a channel. This is the probability that the observed balance is equal to the true balance in that channel. Utility defined like this "equals the minimum probability, over all paths and edges in paths, of a public edge balance equaling the true edge balance after a transaction passes through it" (Tang et al. 2020). In our case, the utility would necessarily be 1 since the observed balance is always equal to the true balance. (Tang et al. 2020) goes on arguing that under certain conditions, their utility metric and the success rate are monotonically related, which leads them to conclude that maximizing utility is as good as maximizing success rate. This is in line with our result that our success rate is not affected by our solution.

b. Optimal Parameters

The bounds derived in these results allow for flexible adjustment in different situations. By lowering the sensitivity Δf and/or the length n of the sequence of transactions we can achieve a lower ϵ and δ. By increasing the capacity of the noise channel we achieve a lower ϵ and δ as well. By increasing t we can trade a higher ϵ for a lower δ. The results mentioned in the introduction were obtained with $t = 200$, $n = 20$, a noise channel with a capacity of 0.12 BTC and an additional capacity of 0.12BTC in the primary channel and a $\Delta f = 2000sat$ resulting in an ϵ of 0.073 and a δ of 0.0021.

A Δf of $2000sat$ might seem insufficient if we look at the average size of a payment in LN. Based on information shared with us by the operator of the lightning.watch node[1], at the time of writing a top 100 node ranked on capacity and a top 10 node ranked on number of channels, we learn that the average payment is $18000sat$. But this need not be a problem. LN allows for splitting up payments in multiple parts, which are sent using a different path for each part. Multipath payments were introduced to increase the likelihood of a successful payment by increasing the likelihood of the individual parts succeeding. We can use this feature to break up our payment in chunks smaller or equal to the Δf, that are sent out over different noised paths, achieving approximate differential privacy even for payments bigger than Δf.

c. Practical Limitations

Even when we choose fairly modest parameters, as we have done in this thesis, we see that the capacity of the noise channel becomes large quickly. With $n = 20, t = 200$ and $\Delta f = 6000sat$ it is impossible for ϵ to be any lower than 0.31 because of the maximum capacity of payment channels (figure 5.3).

The LN protocol supports the feature `option_support_large_channel` which allows for channels with a capacity greater than or equal to $2^{24}sat$. But both peers have to support this feature for the funder to open up a large channel between them, so this option is not always available.

[1] public key: 03e691f81f08c56fa876cc4ef5c9e8b727bd682cf35605be25d48607a802526053

Furthermore, there is no way of knowing whether a relay along a route of ones choosing noises the payment or not. And with the capital costs being as they are, there is an incentive for *not* noising payments. So one can only be sure of payments being noised in channels one controls. Recently techniques requiring trusted execution environments with trusted attestation (Lind et al. 2019) have been proposed that could be used for enforcing correct noising of payments. But this would have a huge impact on the LN protocol.

Our strong privacy guarantees only hold for a fixed number of transactions. After this, the noise channel should be either replaced by a new one (*channel churn*) or rebalanced to start again at a balance of $\frac{a}{2}$. A third option is to use services like *Loop In* and *Loop Out*[2] provided by Lightning Labs, the company behind the LN client LND, to manage the balance of the noise channel. The benefit of channel churn is that the closing transaction can not be linked to the private channel, so no information can be leaked about the amount of noise generated so if the operator replaces the noise channel with a new one every n transactions, the approximate differential privacy is retained for streams of arbitrary length. Rebalancing and *Loops* have the benefit of saving on on-chain transaction fees but run the risk of leaking the amount of noise generated for the previous n transactions when applied naively. Optimizing noise channel rebalancing is a topic for further research.

6.3.2 Payment Splitting and Switching

Our results show that PSS can greatly reduce the extraction of balance information from the BDA. It can be (part of) a mitigation strategy that impedes potential attackers from mounting a BDA.

a. Limitations

Network simulations like the Lightning Network probing simulator have certain limitations, amongst which are accuracy and validation (Rampfl 2013). The simulator has not been

[2]https://lightning.engineering/loop/

validated and certain aspects of the network are absent from the simulation, e.g. connectivity of nodes, routing policies, balance shifting between probes, in-flight payments and other factors (Biryukov, Naumenko, and Tikhomirov 2022). We also do not take into account LN routing fees.

b. Knowledge of PSS being employed

In our analysis, we assume the attacker has full knowledge of whether PSS is employed. But, the way we have set up our plugin, the plugin announces its support for PSS at `init`, which means it only shares its support with direct peers. The attacker does not know for sure whether PSS is employed or not. This makes for an extra factor of uncertainty. The only way for an attacker to know whether PSS is supported for a specific channel is to connect with both peers on either side of the channel and see if they both support PSS. Even then an attacker does not know whether splitting is used, because the forwarding node is free to use PSS fully, partly or not at all. Although we have confirmed the feasibility of payment splitting with a plugin, we are agnostic as to whether PSS should be an optional addition by the use of a plugin or an integral part of the LN protocol. Even if it were to be an integral part of the LN protocol, in practice an attacker will not have full knowledge of the usage of PSS.

c. Overlapping hops in alternative routes

A helpful mental representation of probing in a PSS context is that instead of probing the balance of a channel, e.g. the balance of a channel between Alice and Bob, the attacker is now probing the total liquidity of Alice in the direction of Bob. At first glance it might seem that an attacker would still be able to monitor liquidity over time like an attacker could with channel balances in the original BDA setting (Herrera-Joancomartí et al. 2019) and deduce individual payments in LN. Overlapping alternative routes make this less tractable than it might seem at first sight. Consider Alice and Carol who have one direct channel between them and one alternative route over a single intermediary hop Bob. Now consider Dave and Carol who also have one direct channel between them and one alternative route over the same intermediary hop Bob. The alternative routes share a

single hop between them, namely the channel between Bob and Carol and the balance in that single hop in the direction of Carol is the lowest balance in both alternative routes (defining the balance of those routes). The direct channels between Alice and Carol and between Dave and Carol are depleted in the direction of Carol (figure 6.1). We assume a powerful adversary that can mount a network-wide BDA between every single payment. If Alice now pays 25 to Carol, this powerful adversary would detect a change in liquidity in 7 out of the 12 possible permutations of nodes (table 6.2). Even in this toy example disambiguating the payment becomes quite hard. Disambiguating the payment becomes completely intractable except for the smallest of graphs and is considered out of scope for our research.

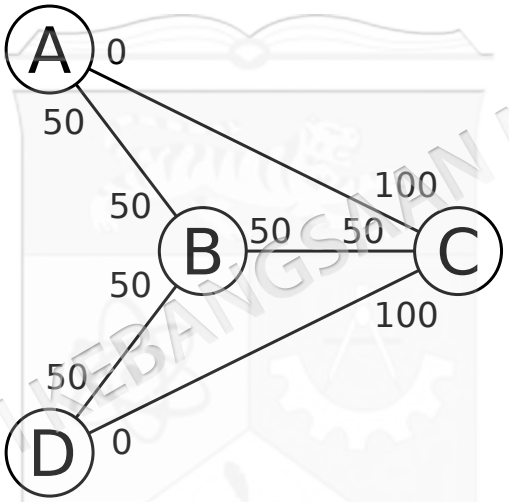

Figure 6.1 Balance distribution at t_0

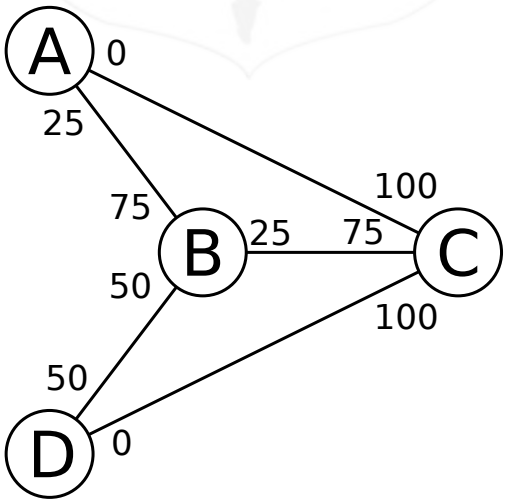

Figure 6.2 Balance distribution at t_1

Table 6.2 Changes in liquidity for all permutations of nodes in a simple graph

liquidity direction	liquidity at t_0	liquidity at t_1
AB	50	25
AC	50	25
AD	50	25
BA	100	100
BC	50	25
BD	100	75
CA	150	175
CB	150	150
CD	150	150
DA	50	50
DB	50	50
DC	50	25

d. PSS as a tool for approximate differentially private payment channels

Since we have also studied the feasibility of approximate differentially private payment channels (DP-channels) in LNit is interesting to see if both countermeasures can be applied together. With the initial approach of DP-channels the actual payment and the noise payment are two separate payments, and a powerful enough adversary could mount a BDA in between those two payments and render the noise payment moot. To prevent this, the node making both payments does not relay any other payments in between those two payments to force the two payments to act as if they were atomic. With PSS we can make the actual payment and the noise payment in parallel as opposed to consecutive, as they are both contingent on the same payment hash. This makes both partial payments atomic as a whole, without requiring to ignore relay requests. It does require a change to the current PSS plugin because the noise payment is circular, meaning that it returns to the sender, and the current plugin does not support that as it was only meant as a proof of concept. In an implementation of PSS that is meant for production this could be one of the requirements.

As we have noted, if multipath payments (Wang et al. 2019; Di Stasi et al. 2018) are supported we can forego negative noise payments by having a payment that consists of a payment over the public channel for the amount of the actual payment minus the noise and a payment over the private channel for the amount of the noise. In this scenario, there is no circular noise payment needed. PSS is a new type of multipath payment that would support exactly this type of scenario. This is an obvious improvement to DP-channels, achievable with PSS, that would reduce the cost of implementing such channels.

It remains an open question whether PSS combined with DP-channels allows for stronger bounds on ϵ, δ)-differential privacy. It is also an open question if PSS would allow for DP-channels without the use of private channels, which would further reduce the cost of implementing DP-channels and improve the feasibility.

e. PSS and enhanced probing.

Enhanced probing is a type of probing where other techniques or aspects of LN are leveraged to improve the results of probing itself (Biryukov, Naumenko, and Tikhomirov 2022). There are two types of enhanced probing: jamming-enhanced probing and policy-aware probing.

Jamming is a denial-of-service attack (EmelyanenkoK 2017; Pérez-Solà et al. 2020). The adversary sends payments to itself or to another node it controls and delays claiming them. In doing so liquidity and payment slots are locked up along the route resulting in congestion. By jamming channels, an attacker can potentially reduce the number of available channels for relaying (or in the case of PSS, available routes) in a hop to one. When we are certain that a payment is relayed through a single channel, a BDA allows for exact disclosure. PSS does not prevent jamming, but it does allow for redirecting partial payments over alternative routes. An attacker would have to take these alternative routes into account and jam them too. Jamming demands considerably more resources when the attacker has to take PSS into account. Also, as noted by Biryukov, Naumenko, and Tikhomirov (2022), this description of jamming makes a few simplifying assumptions. The biggest assumption being made is that an attacker can jam specific

channels, and leave a single particular channel unjammed. In practice, there is no way to achieve that, since any node is free to decide which channel (or route) to use for relaying a payment. So jamming-enhanced probing is more difficult than is obvious at first glance to begin with and PSS adds significantly to this difficulty.

Policy-aware probing is a type of probing that tries to target specific channels by tuning certain parameters such as timeouts and fees. For instance, fee-aware probing, which is a specific kind of policy-aware probing, offers fees for relaying such that only the channel that accepts the lowest fees can forward the payment. This only works if the fee requirements of the channels in a hop are set differently. PSS does not prevent this type of enhanced probing but again, PSS does require the attacker to consider all the alternative routes, making it less likely that a single, specific channel can be targeted.

f. Computational cost

Introducing A. Barvinok's algorithm in the BDA algorithm increases the Big-O time complexity. The original BDA is a binary search with a time complexity of $O(\log n)$. A. Barvinok's algorithm bumps the time complexity up to $O(n^c)$. This is why we had to limit the routes in a routing-hop to a maximum of 9, to keep the running time of the simulation manageable at roughly four hours on an Intel Core i7-8850H 2.60GHz CPU running Ubuntu Linux. During a simulation 5000 hops are targetted. The total amount of hops in the LN snapshot used is 74022. Assuming PSS in the entire LN network, this would bring the runtime of a network-wide BDA on similar hardware to nearly 60 hours. This makes advanced methods such as payment inference unfeasible. The payment discovery algorithm described by Kappos et al. (2021) does not report on the success rate for network balance snapshots spaced 60 hours apart, but snapshots with an interval of 32,768 seconds (over 9 hours) allow for a success rate of less than 5% for revealing payments. This is assuming perfect balance snapshots, which is impossible to begin with if PSS is employed throughout LN.

g. Additional benefits of PSS

Although we have developed and analyzed PSS as a mitigation strategy against BDA, it has side effects that could be considered beneficial to other aspects of LN. The aforementioned rerouting of payments by intermediary hops is something that is explored in the Bailout rerouting protocol (Ersoy, Moreno-Sanchez, and Roos 2023). But where this protocol unlocks and reroutes payments before the payment is completed but after the coins are locked, the rerouting in PSS is only possible before the actual locking of the coins. This makes PSS a less strong mitigation against jamming attacks, but a useful one nonetheless and one that is possible with LN today.

Spider (Sivaraman et al. 2020) is a multi-path transport protocol that utilizes a packet-switched architecture. As we have mentioned previously, PSS turns the circuit-switched nature of LN into a packet-switched one, albeit less comprehensive than Spider does. Nonetheless, network throughput will improve if PSS is implemented, similar to other multi-path payment methods (Osuntokun 2018; Piatkivskyi and Nowostawski 2018).

6.4 EFFECTS ON PRIVACY LOSS

Our third research objective set out to quantify privacy loss before and after the use of our proposed countermeasures in terms of ε-differential privacy (Dwork 2006), channel information coefficient (Tikhomirov et al. 2020) and information gain (Biryukov, Naumenko, and Tikhomirov 2022).

6.4.1 ε-differential privacy

With regard to approximate differential privacy we obtained hard bounds on the privacy loss inflicted by a powerful enough adversary able to mount network-wide BDA's continually. With an escrow of $0.24BTC$ we can hide payments of $2000sat$ and obtain an ϵ of 0.073 and a δ of 0.0021. Since this is the first time differential privacy was deployed

in LN we cannot compare this to previous work. Put differently, that means that currently, without differential privacy, a powerful enough adversary has full certainty whether a payment as actually occurred or not.

6.4.2 Channel Information Coefficient

Although our research does prove that it is infeasible to detect payments when noising is applied, that does not render BDA's completely ineffective. BDA's can still be used to detect larger trends of funds shifting through the network. So it is interesting to see the overall impact on BDA effectiveness.

The impact on the CIC is minimal. The intuition is that any reasonable large Δf requires so much noise padding that a BDA would detect the balance with high precision relative to the capacity of the channel.

6.4.3 Information Gain

The findings with regard to information gain are markedly clear for both direct and indirect probing techniques. Utilizing PSS, the informational advantage gleaned from direct probing of a single channel hop is halved, decreasing from 0.98 to 0.49 ($p < 0.0001$), and for indirect probing, it diminishes from 0.91 to 0.35 ($p < 0.0001$). As the count of parallel channels in the targeted hop grows, the informational gain diminishes for both non-PSS and PSS methods of probing, so the effect of PSS becomes comparatively lesser, but the disparity remains sharp and statistically significant. Specifically, with PSS, the informational advantage of directly probing a 5-channel hop lessens from 0.27 to 0.22 ($p < 0.0001$), and for indirect probing, it reduces from 0.16 to 0.13 ($p < 0.0001$).

6.5 SUMMARY

In this chapter we interpreted the significance of our findings. Our research showed that the two-way BDA algorithm could disclose even more channel balances than earlier work had suggested. With 98.37% of all channels disclosable, nearly no channels remain immune to this type of attack. Furthermore, countermeasures proposed in earlier work are either easily circumvented or come at great cost to the utility of the network.

Our results suggest that two new countermeasures, approximate differential privacy and payment splitting and switching, show promise as a mitigation against the BDA. The former is the first application of the concept of differential privacy in the setting of LN and is able to provably hide payments of reasonable size. The latter is the most successful countermeasure to date when it comes to information gained from a BDA by the attacker.

These effect of these countermeasures on the privacy of the user are quantified using ε-differential privacy, channel information coefficient and information gain respectively.

CHAPTER VII

CONCLUSION AND FUTURE WORK

7.1 OVERVIEW

This final chapter synthesizes the research findings and articulates directions for subsequent inquiry. As delineated in figure 1.2, the investigation commenced with an initial theoretical exploration, spanning two chapters. The inaugural chapter outlined the research subject and objectives, while the subsequent chapter conducted an extensive literature survey, addressing the state of knowledge of LN technology and its challenges. Chapter 3 delineated the research methodology and the blueprint for its application. Chapter 4 described the layed out the empirical outcomes and the performance evaluation, and chapter 6 expanded on this by interpreting and describing the significance of our findings. In this concluding chapter, we synthesize our findings and propose recommendations for future research endeavors.

7.2 CONTRIBUTIONS

Our contributions have effectively addressed the stated research objectives (See section 1.4), as detailed below:

1. Evaluate the effectiveness of BDAs: We extended the BDA with a two-way BDA algorithm, which increased the amount of disclosable channel balances by 9.27 percentage points, confirming the baseline effectiveness of earlier work(See table 6.1).
2. Develop and propose novel mitigation strategies: We have introduced two

countermeasures to the BDA:

1. Approximate differential privacy: This method, discussed in section 3.4, offers strong privacy guarantees but the impact on the channel information coefficient is relatively small (see figure 5.6).
2. Payment Splitting and Switching (PSS): Detailed in Section 3.4.2, this countermeasure significantly reduces the attacker's information gain by up to 62% (see figure 5.10).

3. Quantify the privacy impact of BDAs on LN by measuring privacy loss: We have assessed the impact of both countermeasures on value privacy and found that strong privacy guarantees can be achieved in the sense of approximate differential privacy (See figure b.) with modest impact on the channel information coefficient (See figure 5.6) with respect to the former countermeasure. The latter countermeasure has a very distinct impact on the information gain, decreasing the information gain of the attacker up to 62% (See section 5.2.4).

These discussions and results align with our research objectives, providing a comprehensive justification for our findings.

Additional contributions that were unforeseen when our research objective were drafted are:

1. We uncovered a new vulnerability called Payment of Death (See section 4.3.3) with the risk of an attacker being able to close down payment channels remotely.
2. We created a new version of the LN probing simulator and developed a version that integrates with the `LattE` software (See section 3.4.2, b.).

7.3 CONCLUSIONS

Our initial research presented an improvement to the algorithm of the original BDA. We showed that by approaching a payment channel from both sides instead of from one side, payment channels with a higher capacity than in the original BDA are now also susceptible to this attack. Firstly this accomplishes our first research objective

(See section 1.4, point 1) of replicating and extending the basic BDA. Secondly, since monitoring balances over time makes it possible to detect payments, it can be used to learn information about payments an adversary isn't part of (Malavolta, Moreno-Sanchez, Kate, Maffei, and Ravi 2017).

We exposed differences in the implementation of the BOLT specification by the main three clients. These differences led us to develop a new attack that closes down remote payment channels. We estimated the proportions of each client in LN, by using self-reported information. Based on these proportions we estimated that this attack can be used to take down an important part of LN's entire network capacity.

In search for countermeasure we proposed a new technique for provably hiding payments in a single payment channel in LN by using differential privacy. Our solution is based on adding or subtracting noise to transactions through an additional transaction sourced from a (set of) private channel(s). It is possible to achieve strong privacy guarantees in the sense of differential privacy with this technique. We implemented this technique as a proof of concept with a plugin for the LN client `Core Lightning`. To achieve these privacy guarantees in practice, we have explored the parameters that one would have to work with and showed that implementing noising is tractable in LN but does require locking up a considerable amount of capital.

Although hiding payments is possible, our technique has a limited effect on detecting larger trends in the flow of funds through the LN. So, depending on the goal of the adversary, our technique is equally limited as mitigation against BDA's in general. This is compounded by the fact that it is currently impossible to enforce the use of our technique in LN. Large changes to the LN protocol would be required to make this possible. This leaves us overall pessimistic about adopting differential privacy in LN.

A second technique, developed by us to mitigate BDA in LN allows intermediary parties in a multi-hop payment to reroute the payment to the next hop over an alternative route. We coined the term Payment Splitting and Switching (PSS) for this technique. We showed the viability of this technique by developing a `Core Lightning` plugin that works with Lightning Network today.

Both techniques are viable countermeasures against BDA and as such accomplish our second research objective of developing mitigations against BDA (See section 1.4, point 2).

In measuring the effects on privacy (See section 1.4, point 3) we established that when using the differential privacy plugin with a sensitivity $\Delta f = 2000sat$ we achieve approximate differential privacy with an ϵ of 0.073 and a δ of 0.0021 (See figure b.). The impact on the channel information coefficient is relatively small with the value not dropping below 0.997, regardless of the sensitivity being used (See figure 5.6).

When measuring the effects on privacy (See section 1.4, point 3) for the second countermeasure we have developed, we demonstrated the effect of PSS on the information gain that can be achieved through a BDA by an attacker in an LN network simulation. The simulations, using a real-world network snapshot, showed very pronounced results for Balance Disclosure Attacks (BDA) employing either direct or remote probing. For direct probing of single channel hops the information gain dropped by 50% and for remote probing it dropped by 62%. For multi-channels hops the information gain decrease becomes relatively smaller because the potential information gain without PSS is already smaller to begin with, but the difference remains distinct and significant.

Secondly, PSS forces an increase in the time complexity of the BDA algorithm from $O(\log n)$ to $O(n^c)$. This makes the use of BDA at a scale needed for network-wide payment discovery unfeasible with off-the-shelf hardware.

The Lightning Network shows tremendous promise to improve Bitcoin's scalability and privacy, but also introduces new types of attacks. With this work we we uncovered previously unknown consequences of the attack but also proposed practical and obtainable countermeasures against on the BDA.

7.3.1 Limitations

Network simulations, such as the LN probing simulator, are subject to specific constraints, including issues of accuracy and validation, as noted by Rampfl (2013). The LN probing

simulator is not validated, and it omits key elements of the network such as node connectivity, routing policies, the redistribution of balances between probes, payments in transit, and other relevant factors, as highlighted by Biryukov, Naumenko, and Tikhomirov (2022). Additionally, our analyses do not consider the routing fees associated with the Lightning Network.

The above also applies to Simverse, but to a lesser extent, since Simverse is not a simulation but a testbed that uses an actual network with nodes using the actual LN software. However, Simverse runs on a single computer where all network elements are dockerized, so key factors such as node connectivity, network delays are not omitted like in the LN probing simulator, but are also not a fair representation of the actual LN network encompassing the entire globe.

With respect to approximate differential privacy, we see that even with fairly modest parameters, the requirements for the capacity of the noise channel become substantial very fast and run into the maximal channel capacity.

Also this thesis does not cover how the LN protocol could enforce the use of BDA countermeasures. At the foundation of LN is an incentive structure, namely the routing fees to be obtained by relaying payments. There is a strong argument to be made that this incentive model also incentivizes the use of BDA countermeasures, since they are likely to increase the amount of payments than can be successfully relayed, but this effect on the LN network needs further research.

7.4 FUTURE WORK

In this thesis we already hinted at potential topics for future research. We identify three general research directions that are of interest to the subject matter of this thesis:

- The further exploration of synergies when countermeasures are deployed simultaneously
- The effect of the proposed countermeasures on other types of attack
- The effect of proposed LN protocol changes on BDA and its countermeasures

7.4.1 Synergies Between Countermeasures

It remains an open question whether a potential enhancement of (ϵ, δ)-differential privacy can be obtained through the integration of PSS with DP-channels. Additionally, the question remains open as to whether PSS can facilitate DP-channels without necessitating private channels. Such a development could lead to a reduction in the implementation costs of DP-channels and enhance the feasibility of differential privacy. However, to establish strong bounds for the purpose of differential privacy it is paramount to mathematically model an LN payment through a routing-hop and the current model does not easily allow for that.

7.4.2 The Effect on Other Attacks

Our results, especially with regards to payment splitting and switching show promise in mitigating other types of attack as well. PSS shows similarities with the Bailout rerouting protocol (Ersoy, Moreno-Sanchez, and Roos 2023) which is a countermeasure against jamming attacks. At first glance PSS seems a less strong candidate, but with the additional benefit of being compatible with the current LN protocol. Future research should quantify to what extent PSS can be used as a countermeasure against jamming attacks.

7.4.3 The Effect of Proposed LN Protocol Changes

The LN protocol as specified through the Basis of Lightning Technology (BOLT) ("Basis of Lightning Technology (BoLT), Lightning Network In-Progress Specifications" 2022) documents is a living protocol that changes over time. Recently proposed changes might have substantial impact on our findings. For example the proposed protocol change for dual-funded channels, where both nodes contribute funds to the funding transaction makes it possible to open a channel that is differentially private from the get-go. Similarly, the proposed changes for Point Time Locked Contracts (PTLCs) would force some parts of our plugins to be reconsidered.

This is a two-way street. Our findings can also inform future protocol changes. With regards to differential privacy for instance, we envisioned a modification to the Lightning Network (LN) protocol, enabling the allocation of a portion of a channel's capacity for the purpose of noise generation. In this revised framework, a channel maintains two distinct balances. The first balance corresponds to the conventional channel balance, akin to the existing model, but with a capacity equal to the funding transaction minus the capacity allocated for noise generation. The second balance is reserved for noise generation. A transaction conducted through this channel would involve two Hash Time Locked Contracts (HTLCs): one to update the primary balance by the amount adjusted for noise, and the other HTLC to update the noise generation balance. This approach bears similarities to the parallel channels strategy but is integrated into the LN protocol itself. Moreover, with the incorporation of differential privacy into the protocol, network peers can mutually consent to establish a differential private payment channel as part of the broader channel parameters agreed upon during channel initialization.

BIBLIOGRAPHY

Albert, Réka, Hawoong Jeong, and Albert-László Barabási. 2000. "Error and Attack Tolerance of Complex Networks." *Nature* 406 (6794): 378–82. https://doi.org/10.1038/35019019.

Androulaki, Elli, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. 2013. "Evaluating User Privacy in Bitcoin." In *International Conference on Financial Cryptography and Data Security*, 34–51. Springer.

Avarikioti, Georgia, Gerrit Janssen, Yuyi Wang, and Roger Wattenhofer. 2018. "Payment Network Design with Fees." In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, edited by Joaquin Garcia-Alfaro, Jordi Herrera-Joancomartí, Giovanni Livraga, and Ruben Rios, 11025:76–84. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-00305-0_6.

Back, Adam, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. n.d. "Enabling Blockchain Innovations with Pegged Sidechains."

Backes, Michael, and Sebastian Meiser. 2014. "Differentially Private Smart Metering with Battery Recharging." In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8247 LNCS:194–212. https://doi.org/10.1007/978-3-642-54568-9_13.

Barber, Simon, Xavier Boyen, Elaine Shi, and Ersin Uzun. 2012. "Bitter to Better—How to Make Bitcoin a Better Currency." In *International Conference on Financial Cryptography and Data Security*, 399–414. Springer.

Barvinok, Alexander I. 1994. "A Polynomial Time Algorithm for Counting Integral Points in Polyhedra When the Dimension Is Fixed." *Mathematics of Operations Research* 19 (4): 769–79. https://doi.org/10.1287/moor.19.4.769.

"Basis of Lightning Technology (BoLT), Lightning Network In-Progress Specifications." 2022. https://github.com/lightning/bolts/blob/master/00-introduction.md.

Bez, Mirko, Giacomo Fornari, and Tullio Vardanega. 2019. "The Scalability Challenge of

Ethereum: An Initial Quantitative Analysis." In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, 167–76. San Francisco East Bay, CA, USA: IEEE. https://doi.org/10.1109/SOSE.2019.00031.

Birman, Ken. 2007. "The Promise, and Limitations, of Gossip Protocols." *ACM SIGOPS Operating Systems Review* 41 (5): 8–13. https://doi.org/10.1145/1317379.1317382.

Biryukov, Alex, Gleb Naumenko, and Sergei Tikhomirov. 2022. "Analysis and Probing of Parallel Channels in the Lightning Network." In *Financial Cryptography and Data Security*. Lecture Notes in Computer Science. Springer Cham.

Bondy, Warren H., and William Zlot. 1976. "The Standard Error of the Mean and the Difference Between Means for Finite Populations." *The American Statistician* 30 (2): 96–97. https://doi.org/10.1080/00031305.1976.10479149.

Camenisch, Jan, and Anna Lysyanskaya. 2006. "A Formal Treatment of Onion Routing." In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/11535218_11.

Chan, T.-H. Hubert, Elaine Shi, and Dawn Song. 2011. "Private and Continual Release of Statistics." *ACM Transactions on Information and System Security* 14 (3): 1–24. https://doi.org/10.1145/2043621.2043626.

Chaudhuri, Kamalika, Jacob Imola, and Ashwin Machanavajjhala. 2019. "Capacity Bounded Differential Privacy." *Advances in Neural Information Processing Systems* 32. https://arxiv.org/abs/1907.02159.

Conti, Mauro, Sandeep Kumar E, Chhagan Lal, and Sushmita Ruj. 2018. "A Survey on Security and Privacy Issues of Bitcoin." *IEEE Communications Surveys and Tutorials* 20 (4): 3416–52. https://doi.org/10.1109/COMST.2018.2842460.

Dandekar, Pranav, Ashish Goel, Ramesh Govindan, and Ian Post. 2011. "Liquidity in Credit Networks: A Little Trust Goes a Long Way." In *Proceedings of the 12th ACM Conference on Electronic Commerce*, 147–56. San Jose California USA: ACM. https://doi.org/10.1145/1993574.1993597.

Danezis, George, and Ian Goldberg. 2009. "Sphinx: A Compact and Provably Secure Mix Format." *Proceedings - IEEE Symposium on Security and Privacy*, 269–82. https://doi.org/10.1109/SP.2009.15.

De Loera, Jesús A., Raymond Hemmecke, Jeremiah Tauzer, and Ruriko Yoshida. 2004.

"Effective Lattice Point Counting in Rational Convex Polytopes." *Journal of Symbolic Computation* 38 (4): 1273–1302. https://doi.org/10.1016/j.jsc.2003.04.003.

Decker, Christian, and Roger Wattenhofer. 2015. "A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels." *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9212: 3–18. https://doi.org/10.1007/978-3-319-21741-3_1.

Di Stasi, Giovanni, Stefano Avallone, Roberto Canonico, and Giorgio Ventre. 2018. "Routing Payments on the Lightning Network." *Proceedings - IEEE 2018 International Congress on Cybermatics: 2018 IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, iThings/Gree*, 1161–70. https://doi.org/10.1109/Cybermatics_2018.2018.00209.

Dmitrienko, Alexandra, David Noack, and Moti Yung. 2017. "Secure Wallet-Assisted Offline Bitcoin Payments with Double-Spender Revocation." In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 520–31. Abu Dhabi United Arab Emirates: ACM. https://doi.org/10.1145/3052973.3052980.

Dwork, Cynthia. 2006. "Differential Privacy." *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4052 LNCS: 1–12. https://doi.org/10.1007/11787006_1.

Dwork, Cynthia, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. 2010. "Differential Privacy Under Continual Observation." In *Proceedings of the 42nd ACM Symposium on Theory of Computing - STOC '10*, 31:715–24. New York, New York, USA: ACM Press. https://doi.org/10.1145/1806689.1806787.

Easley, David, Maureen O'Hara, and Soumya Basu. 2019. "From Mining to Markets: The Evolution of Bitcoin Transaction Fees." *Journal of Financial Economics* 134 (1): 91–109. https://doi.org/10.1016/j.jfineco.2019.03.004.

EmelyanenkoK. 2017. "Payment Channel Congestion via Spam-Attack · Issue #182 · Lightning/Bolts." *GitHub*. https://github.com/lightning/bolts/issues/182.

Ersoy, Oguzhan, Pedro Moreno-Sanchez, and Stefanie Roos. 2023. "Get Me Out of This Payment! Bailout: An HTLC Re-routing Protocol." In *Financial Cryptography and Data Security*.

Fortunato, Santo, and Claudio Castellano. 2012. "Community Structure in Graphs." In

Computational Complexity, edited by Robert A. Meyers, 490–512. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-1800-9_33.

Gabizon, Ariel, Zachary J. Williamson, and Oana Ciobotaru. 2019. "PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive Arguments of Knowledge."

Gangwal, Ankit, Haripriya Ravali Gangavalli, and Apoorva Thirupathi. 2023. "A Survey of Layer-two Blockchain Protocols." *Journal of Network and Computer Applications* 209 (January): 103539. https://doi.org/10.1016/j.jnca.2022.103539.

Garcia-Alfaro, Joaquin, Jordi Herrera-Joancomartí, Emil Lupu, Joachim Posegga, Alessandro Aldini, Fabio Martinelli, and Neeraj Suri. 2015. *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance*. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 8872. https://doi.org/10.1007/978-3-319-17016-9.

Girvan, M., and M. E. J. Newman. 2002. "Community Structure in Social and Biological Networks." *Proceedings of the National Academy of Sciences* 99 (12): 7821–26. https://doi.org/10.1073/pnas.122653799.

Green, Matthew, and Ian Miers. 2017. "Bolt: Anonymous Payment Channels for Decentralized Currencies." In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS '17*, 473–89. New York, New York, USA: ACM Press. https://doi.org/10.1145/3133956.3134093.

Harris, Jona, and Aviv Zohar. 2020. "Flood & Loot: A Systemic Attack On The Lightning Network." https://arxiv.org/abs/2006.08513.

Hearn, Mike. 2013. "Micro-Payment Channels Implementation Now in Bitcoinj." *Bitcointalk. Org*.

Herrera-Joancomartí, Jordi, Guillermo Navarro-Arribas, Alejandro Ranchal-Pedrosa, Cristina Pérez-Solà, and Joaquin Garcia-Alfaro. 2019. "On the Difficulty of Hiding the Balance of Lightning Network Channels," 602–12. https://doi.org/10.1145/3321705.3329812.

Hu, Kexin, and Zhenfeng Zhang. 2018. "Fast Lottery-Based Micropayments for Decentralized Currencies." In *Information Security and Privacy*, edited by Willy Susilo and Guomin Yang, 10946:669–86. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-93638-3_38.

Huang, Chengpeng, Rui Song, Shang Gao, Yu Guo, and Bin Xiao. 2024. "Data

Availability and Decentralization: New Techniques for Zk-Rollups in Layer 2 Blockchain Networks." arXiv. https://doi.org/10.48550/arXiv.2403.10828.

Kappos, George, Haaroon Yousaf, Ania Piotrowska, Sanket Kanjalkar, Sergi Delgado-Segura, Andrew Miller, and Sarah Meiklejohn. 2021. "An Empirical Analysis of Privacy in the Lightning Network." In *Financial Cryptography and Data Security*, edited by Nikita Borisov and Claudia Diaz, 167–86. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-64322-8_8.

Khalil, Rami, and Arthur Gervais. 2017. "Revive: Rebalancing Off-Blockchain Payment Networks." *Proceedings of the ACM Conference on Computer and Communications Security*, 439–53. https://doi.org/10.1145/3133956.3134033.

Koshy, Philip, Diana Koshy, and Patrick McDaniel. 2014. "An Analysis of Anonymity in Bitcoin Using P2p Network Traffic." In *International Conference on Financial Cryptography and Data Security*, 469–85. Springer.

Kumble, Satwik Prabhu, Dick Epema, and Stefanie Roos. 2023. "Game-Theoretic Analysis of (Non-)Refundable Fees in the Lightning Network." https://doi.org/10.48550/ARXIV.2310.04058.

Lamport, Leslie, Robert Shostak, and Marshall Pease. 1982. "The Byzantine Generals Problem." *ACM Transactions on Programming Languages and Systems* 4 (3): 382–401. https://doi.org/10.1145/357172.357176.

Langenheldt, Christian, Mikkel Alexander Harlev, Hao Hua, Sun Yin, Raghava Rao Mukkamala, and Ravi Vatrapu. 2018. "Breaking Bad : De-Anonymising Entity Types on the Bitcoin Blockchain Using Supervised Machine Learning." In, 9:3497–3506. https://doi.org/10.24251/HICSS.2018.443.

Lee, Seungjin, and Hyoungshick Kim. 2020. "On the Robustness of Lightning Network in Bitcoin." *Pervasive and Mobile Computing* 61 (January): 101108. https://doi.org/10.1016/j.pmcj.2019.101108.

Leinweber, Marc, Matthias Grundmann, Leonard Schönborn, and Hannes Hartenstein. 2019. "TEE-Based Distributed Watchtowers for Fraud Protection in the Lightning Network." In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, edited by Cristina Pérez-Solà, Guillermo Navarro-Arribas, Alex Biryukov, and Joaquin Garcia-Alfaro, 11737:177–94. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-31500-9_11.

Lind, Joshua, Oded Naor, Ittay Eyal, Florian Kelbert, Emin Gün Sirer, and Peter Pietzuch. 2019. "Teechain: A Secure Payment Network with Asynchronous Blockchain Access." In *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, 63–79. New York, NY, USA: ACM. https://doi.org/10.1145/3341301.3359627.

Liu, Bowen, Pawel Szalachowski, and Siwei Sun. 2020. "Fail-Safe Watchtowers and Short-lived Assertions for Payment Channels." In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 506–18. Taipei Taiwan: ACM. https://doi.org/10.1145/3320269.3384716.

Malavolta, Giulio, Pedro Moreno-Sanchez, Aniket Kate, and Matteo Maffei. 2017. "SilentWhispers: Enforcing Security and Privacy in Decentralized Credit Networks." In *Proceedings 2017 Network and Distributed System Security Symposium*. March. Reston, VA: Internet Society. https://doi.org/10.14722/ndss.2017.23448.

Malavolta, Giulio, Pedro Moreno-Sanchez, Aniket Kate, Matteo Maffei, and Srivatsan Ravi. 2017. "Concurrency and Privacy with Payment-Channel Networks." In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS '17*, 455–71. New York, New York, USA: ACM Press. https://doi.org/10.1145/3133956.3134096.

Malavolta, Giulio, Pedro Moreno-Sanchez, Clara Schneidewind, Aniket Kate, and Matteo Maffei. 2019. "Anonymous Multi-Hop Locks for Blockchain Scalability and Interoperability." In *Proceedings 2019 Network and Distributed System Security Symposium*. Reston, VA: Internet Society. https://doi.org/10.14722/ndss.2019.23330.

Maxwell, Gregory, Andrew Poelstra, Yannick Seurin, and Pieter Wuille. 2019. "Simple Schnorr Multi-Signatures with Applications to Bitcoin." *Designs, Codes and Cryptography* 87 (9): 2139–64. https://doi.org/10.1007/s10623-019-00608-x.

Mazumdar, Subhra, Prabal Banerjee, and Sushmita Ruj. 2020. "Time Is Money: Countering Griefing Attack in Lightning Network." In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 1036–43. https://doi.org/10.1109/TrustCom50675.2020.00138.

Meiklejohn, Sarah, and Claudio Orlandi. 2015. "Privacy-Enhancing Overlays in Bitcoin." *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-662-48051-9_10.

Meiklejohn, Sarah, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. 2013. "A Fistful of Bitcoins: Characterizing Payments Among Men with No Names." *Communications of the ACM* 59 (4): 86–93. https://doi.org/10.1145/2896384.

Miller, Andrew, Iddo Bentov, Surya Bakshi, Ranjit Kumaresan, and Patrick McCorry. 2019. "Sprites and State Channels: Payment Networks That Go Faster Than Lightning." In *Financial Cryptography and Data Security*, edited by Ian Goldberg and Tyler Moore, 11598:508–26. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-32101-7_30.

Mizrahi, Ayelet, and Aviv Zohar. 2020. "Congestion Attacks in Payment Channel Networks." https://arxiv.org/abs/2002.06564.

Nagaraja, Shishir. 2007. "Topology of Covert Conflict: (Transcript of Discussion)." In *Security Protocols*, edited by David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, et al., 4631:329–32. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-77156-2_41.

Nakamoto, Satoshi. 2008. "Bitcoin: A Peer-to-Peer Electronic Cash System."

Network-Fast, Raiden. 2018. "Cheap, Scalable Token Transfers for Ethereum." *Accessed: Jul*. https://raiden.network/.

Osuntokun, Olaoluwa. 2018. "[Lightning-dev] AMP: Atomic Multi-Path Payments over Lightning."

Pass, Rafael, and Abhi Shelat. 2015. "Micropayments for Decentralized Currencies." In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 207–18. Denver Colorado USA: ACM. https://doi.org/10.1145/2810103.2813713.

Pérez-Solà, Cristina, Alejandro Ranchal-Pedrosa, Jordi Herrera-Joancomartí, Guillermo Navarro-Arribas, and Joaquin Garcia-Alfaro. 2020. "LockDown: Balance Availability Attack Against Lightning Network Channels." In, 245–63. https://doi.org/10.1007/978-3-030-51280-4_14.

Piatkivskyi, Dmytro, and Mariusz Nowostawski. 2018. "Split Payments in Payment Networks." In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, edited by Joaquin Garcia-Alfaro, Jordi Herrera-Joancomartí, Giovanni Livraga, and

Ruben Rios, 11025:67–75. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-00305-0_5.

Poon, Joseph, and Vitalik Buterin. 2017. "Plasma: Scalable Autonomous Smart Contracts." *White Paper*, 1–47.

Poon, Joseph, and Thaddeus Dryja. 2016. "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments." 3. Vol. 18.

Porter, Mason A., Jukka-Pekka Onnela, and Peter J. Mucha. 2009. "Communities in Networks." *Notices of the AMS* 56 (9). https://doi.org/10.48550/ARXIV.0902.3788.

Rahimpour, Sonbol, and Majid Khabbazian. 2022. "Torrent: Strong, Fast Balance Discovery in the Lightning Network." In *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 1–7. Shanghai, China: IEEE. https://doi.org/10.1109/ICBC54727.2022.9805520.

Rampfl, Sebastian. 2013. "Network Simulation and Its Limitations." https://doi.org/10.2313/NET-2013-08-1_08.

Rebello, Gabriel Antonio F., Gustavo F. Camilo, Lucas Airam C. de Souza, Maria Potop-Butucaru, Marcelo Dias de Amorim, Miguel Elias M. Campista, and Luís Henrique M. K. Costa. 2024. "A Survey on Blockchain Scalability: From Hardware to Layer-Two Protocols." *IEEE Communications Surveys & Tutorials*, 1–1. https://doi.org/10.1109/COMST.2024.3376252.

Reed, Michael G., Paul F. Syverson, and David M. Goldschlag. 1998. "Anonymous Connections and Onion Routing." *IEEE Journal on Selected Areas in Communications* 16 (4): 482–93. https://doi.org/10.1109/49.668972.

Reid, Fergal, and Martin Harrigan. 2013. "An Analysis of Anonymity in the Bitcoin System." In *Security and Privacy in Social Networks*, 197–223. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-4139-7_10.

Roos, Stefanie, Martin Beck, and Thorsten Strufe. 2016. "Anonymous Addresses for Efficient and Resilient Routing in F2F Overlays." In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 1–9. San Francisco, CA, USA: IEEE. https://doi.org/10.1109/INFOCOM.2016.7524553.

Roos, Stefanie, Pedro Moreno-Sanchez, Aniket Kate, and Ian Goldberg. 2018. "Settling Payments Fast and Private: Efficient Decentralized Routing for Path-Based Transactions." In *Proceedings 2018 Network and Distributed System Security Symposium*. Reston,

VA: Internet Society. https://doi.org/10.14722/ndss.2018.23252.

Rosenfeld, Meni. 2014. "Analysis of Hashrate-Based Double Spending." arXiv. https://arxiv.org/abs/1402.2009.

Sguanci, Cosimo, Roberto Spatafora, and Andrea Mario Vergani. 2021. "Layer 2 Blockchain Scaling: A Survey." arXiv. https://doi.org/10.48550/arXiv.2107.10881.

Shikhelman, Clara, and Sergei Tikhomirov. 2022. "Unjamming Lightning: A Systematic Approach."

Singh, Amritraj, Kelly Click, Reza M. Parizi, Qi Zhang, Ali Dehghantanha, and Kim-Kwang Raymond Choo. 2020. "Sidechain Technologies in Blockchain Networks: An Examination and State-of-the-Art Review." *Journal of Network and Computer Applications* 149 (January): 102471. https://doi.org/10.1016/j.jnca.2019.102471.

Sivaraman, Vibhaalakshmi, Shaileshh Bojja Venkatakrishnan, Mohammad Alizadeh, Giulia Fanti, and Pramod Viswanath. 2018. "Routing Cryptocurrency with the Spider Network." In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, 29–35. New York, NY, USA: ACM. https://doi.org/10.1145/3286062.3286067.

Sivaraman, Vibhaalakshmi, Shaileshh Bojja Venkatakrishnan, Kathleen Ruan, Parimarjan Negi, Lei Yang, Radhika Mittal, Giulia Fanti, and Mohammad Alizadeh. 2020. "High Throughput Cryptocurrency Routing in Payment Channel Networks."

Sompolinsky, Yonatan, and Aviv Zohar. 2015. "Secure High-Rate Transaction Processing in Bitcoin (Full Version)." *Financial Cryptography and Data Security - 19th International Conference, FC 2015*, 507–27.

Spagnuolo, Michele, Federico Maggi, and Stefano Zanero. 2014. "Bitiodine: Extracting Intelligence from the Bitcoin Network." In *International Conference on Financial Cryptography and Data Security*, 457–68. Springer.

Tadge Dryja. 2016. "Unlinkable-Outsourced-Channel-Monitoring." https://diyhpl.us/wiki/transcripts/sca
outsourced-channel-monitoring/.

Tang, Weizhao, Weina Wang, Giulia Fanti, and Sewoong Oh. 2020. "Privacy-Utility Tradeoffs in Routing Cryptocurrency over Payment Channel Networks." In *Abstracts of the 2020 SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems*, 81–82. SIGMETRICS '20. Boston, MA, USA: Association for Computing Machinery. https://doi.org/10.1145/3393691.3394213.

"The State of Lightning: Volume 2." 2022. Arcane Research.

Thibault, Louis Tremblay, Tom Sarry, and Abdelhakim Senhaji Hafid. 2022. "Blockchain Scaling Using Rollups: A Comprehensive Survey." *IEEE Access* 10: 93039–54. https://doi.org/10.1109/ACCESS.2022.3200051.

Tikhomirov, Sergei, Rene Pickhardt, Alex Biryukov, and Mariusz Nowostawski. 2020. "Probing Channel Balances in the Lightning Network," no. 1. https://arxiv.org/abs/2004.00333.

Tsuchiya, P. F. 1988. "The Landmark Hierarchy: A New Hierarchy for Routing in Very Large Networks." *ACM SIGCOMM Computer Communication Review* 18 (4): 35–42. https://doi.org/10.1145/52325.52329.

Wang, Peng, Hong Xu, Xin Jin, and Tao Wang. 2019. "Flash." In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, 370–81. November 2018. New York, NY, USA: ACM. https://doi.org/10.1145/3359989.3365411.

Zhao, Zhixin, Chunpeng Ge, Lu Zhou, and Huaqun Wang. 2021. "Fully Discover the Balance of Lightning Network Payment Channels." In *Wireless Algorithms, Systems, and Applications*, edited by Zhe Liu, Fan Wu, and Sajal K. Das, 12937:159–73. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-85928-2_13.

Zhao, Zhixin, Lu Zhou, and Chunhua Su. 2021. "Systematic Research on Technology and Challenges of Lightning Network." In *2021 IEEE Conference on Dependable and Secure Computing (DSC)*, 1–8. https://doi.org/10.1109/DSC49826.2021.9346275.

APPENDIX A

PROOFS

Proof for Lemma 4. The difference between $\mathbb{C}$ and $\hat{c}_b$ lies in the fact that $\mathbb{C}$ keeps track of the balance bal of the noise generating channel and cuts the noise entirely if it exceeds the capacity of the channel. Given a transaction stream σ of length n and a set S,

$$
\begin{aligned}
&|Pr[\mathbb{C}(\sigma) \in S] - Pr[\hat{c}_b(\sigma) \in S]| \\
&\leq Pr\left[\exists k \in 1, ...n. \mathbb{C}\Big(\sigma(k)\Big) \neq \hat{c}_b\Big(\sigma(k)\Big)\right] \\
&\leq Pr\left[\exists k \in 1, ...n. \left|\sum_{j=1}^{k}\left(\hat{c}_b\Big(\sigma(k)\Big) - c\Big(\sigma(k)\Big)\right)\right| > \frac{a}{2}\right]
\end{aligned}
$$

□

Proof for Lemma 5. $Lap(\frac{1}{\lambda})$ has the following probability density function.

$$
Lap(s, x) = \frac{\lambda}{2}\begin{cases} e^{x\lambda} & \text{if } x < 0 \\ e^{-x\lambda} & \text{if } x \geq 0 \end{cases}
$$

Taking the absolute value gives the following probability density function:

$$
|Lap(s, x)| = \lambda e^{-x\lambda}
$$

Given $\lambda = \frac{\epsilon}{\Delta f}$ which has an expected value of $\frac{\Delta f}{\epsilon}$, the expected value of the sum is

$$
E\left[\sum^{n}\left|Lap\left(\frac{\Delta f}{\epsilon}\right)\right|\right] = n \cdot \frac{\Delta f}{\epsilon}.
$$

The variance of $\left|Lap\left(\frac{\Delta f}{\epsilon}\right)\right|$ is $2 \cdot \left(\frac{\Delta f}{\epsilon}\right)^2$ and since the individual random variables are

uncorrelated the sum becomes:

$$var\left(\sum^{n}\left|Lap\left(\frac{\Delta f}{\epsilon}\right)\right|\right) = 2n \cdot \left(\frac{\Delta f}{\epsilon}\right)^2.$$

We now abbreviate $\sum^{n}\left|Lap\left(\frac{\Delta f}{\epsilon}\right)\right|$ with X_n for clarity.

$$\begin{aligned}
&Pr\left[X_n \geq (n+t)\cdot\frac{\Delta f}{\epsilon}\right]\\
\leq &Pr\left[\left|X_n - n\cdot\frac{\Delta f}{\epsilon}\right| \geq t\cdot\frac{\Delta f}{\epsilon}\right]\\
= &Pr\left[|X_n - E[X_n]| \geq t\cdot\frac{\Delta f}{\epsilon}\right]\\
\leq &\frac{var(X_n)}{\left(t\cdot\frac{\Delta f}{\epsilon}\right)^2}\\
\leq &\frac{2n\cdot\left(\frac{\Delta f}{\epsilon}\right)^2}{\left(t\cdot\frac{\Delta f}{\epsilon}\right)^2} = \frac{2n}{t^2}
\end{aligned}$$

□

Proof for Theorem 1. By assumption, $\hat{c}$ is $(\epsilon, 0)$-differentially private. By lemma 1, $\hat{c}_b$ is $(\epsilon, (e^{\epsilon}+1)\cdot P_b)$-differentially private, where P_b is the statistical distance between $\hat{c}$ and $\hat{c}_b$. By applying lemma 1 again we obtain that $\mathbb{C}$ is $(\epsilon, (e^{\epsilon}+1)\cdot(P_b + P_c))$-differentially private, where P_c is the statistical distance between $\hat{c}_b$ and $\mathbb{C}$. □

Lemma 6. *Let c be an LN channel with capacity b and c' be an LN channel with capacity b'. Let bal_{ab_c} and bal_{ba_c} be the balance on the left and right side respectively of channel c, similarly for $bal_{ab_{c'}}$ and $bal_{ba_{c'}}$ w.r.t channel c'. Assume that $bal_{ab_{c'}} \geq bal_{ab_c}$ and $bal_{ba_{c'}} \geq bal_{ba_c}$ and $b' = b + a$. Let $T = t_1, ..., t_n$ be a sequence of transactions that is approved in c where $t_i \in \mathbb{Z} \setminus \{0\}$. When $t_i > 0$ the transaction is assumed to move from left to right, likewise when $t_i < 0$ it is assumed to move from right to left. Assume, for sake of contradiction, that there exists a transaction $t_i \in T$ that is rejected in c'. If $t_i > 0$ then $bal_{ab_{c'}} - \sum_{j=1}^{i-1} t_j \geq 0$ but $bal_{ab_{c'}} - \sum_{j=1}^{i} t_j < 0$. However, since t_i was approved in c it holds that $bal_{ab_c} - \sum_{j=1}^{i} t_j \geq 0$, but it also holds that $bal_{ab_{c'}} \geq bal_{ab_c}$ which is a contradiction. The proof for the case where $t_i > 0$ is identical. Therefor any sequence that is approved in c will be approved in c'.*

Proof for Theorem 2. From lemma 6 we know that if we add a capacity to a channel, that every sequence of transactions that would be approved without the added capacity will be approved with the added capacity given the assumption that the balances on either side are at least equal to the balances before adding the capacity. If the same sequence of transactions were to be noised, than that sequence of transactions would be approved in the channel with added capacity if the total amount of added noise is not bigger than a. Therefor in order to proof the theorem it suffices to show that the the total amount of noise cannot be larger than a. If we set the added capacity a to be the capacity of the noise channel and bal_{ab} and bal_{ba} are both increased with $\frac{a}{2}$ at initialisation then this is true by definition. □